Diciembre 1985-300 ptas.

Todospectrum

AÑO 2.-NUMERO 16.

REVISTA EXCLUSIVA PARA USUARIOS

Hardware práctico:

Monta tu propio circuito

Cálculos con infinita precisión

Un super juego para teclear:

La última batalla



EL REGALO DE ESTAS FIESTAS QUE VALE POR TODOS

SPECTRUM PLUS Y 128 ¡ALLELUIA, ALLELUIA!

Le presentamos el regalo de estas Navidades que vale por todos.

Si está pensando en regalar juegos, futuro, aprobado en Matemáticas... regale Spectrum.

La familia de ordenadores familiares más vendida del mundo.

Y la gran novedad del mercado: Spectrum 128 K. Una exclusiva mundial con teclado en español, y teclado adicional para editar programas, textos, controlar juegos o como calculadora.

Con un simple comando puede convertirse en Spectrum Plus. Dos ordenadores en uno solo.

Y una potente memoria RAM de 128 K que le permite ejecutar los programas más complicados, almacenar más información...

SPECTRUM PLUS Y 128
ORDENADORES CON BUENA ESTRELLA

DISTRIBUIDOR EXCLUSIVO

investronica



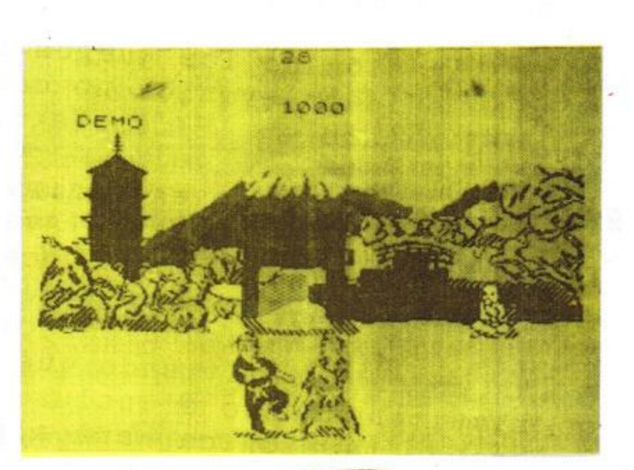
Tomás Bretón, 60. Telf. (91) 467 82 10. Télex 23399 IYCO E. 28045 Madrid
Camp, 80. Telf. (93) 211 26 58-211 27 54. 08022 Barcelona





AÑO II





Con el mes de diciembre viene un paréntesis de vacaciones que todo el mundo aprovecha para descansar y dedicarse a sus distracciones favoritas. Para los lectores de nuestra revista ofrecemos una opción muy interesante, ya que incluimos en este número un juego de estrategia con un acabado profesional. Debido a la extensión de este juego hemos tenido que suprimir la serie de código máquina que seguirá en el número de enero.

Los usuarios de QL también van a poder teclear otro programa, ya que incluimos el primero enviado por un lector, aunque esperamos que nos lleguen muchos otros próximamente. Muchas gracias por no faltar a la cita con nosotros y esperamos que nos sigáis leyendo el año que viene. Feliz 1986.

ACTUALIDAD

```
10 REM La noticia, que ya se venia rumoreando, ha caido como
       una bomba al confirmarse este pasado SIMO. Investronica,
       importador oficial de los productos Sinclair para
       España, ha cogido la representación de los nuevos Atari
       sin abandonar Spectrum o QL. Tal como están las cosas,
       esto significa que ahora tiene dos máquinas (QL y Atari
       520) que se hacen la competencia mutua.

20 REM Sinclair no solo no se hunde (como pretendian muchos),
       si no que se anuncian y rumorean nuevos productos, entre
       ellos una versión mejorada del QL. Con el nombre de
       ENIGMA, tendrá 1 megabyte de memoria, un 68000 (no un
       68008 como el modelo actual), disco de 3 pulgadas y
       media y el sistema operativo basado en iconos y ratón.

30 REM La invasión de los juegos basados en series famosas del
       cine o la televisión no cesa. El último que se anuncia
       está basado en 'El coche fantástico'.

40 REM En Inglaterra se encuentra de moda el 'MUD', abreviatura
       de Multi User Dungeon (Aventura Multiusuario). Esto no
       es más que un ordenador central propiedad de la British
       Telecom (la compañia telefónica británica) al que los
       jugadores se pueden conectar por medio de un teléfono y
       un modem, permitiendo que varias personas juegen
       simultaneamente e incluso se peleen entre si. ¿Para
       cuando telefónica hará algo igual?.
```

CINCO HORAS CON SCREEN$

Cuando se compra un ordenador como el Spectrum, uno siempre piensa que va a escribir sus propios programas de juegos. Pero en seguida cae en la cuenta de que la decisión necesariamente implica pasar por comprender el funcionamiento de la zona de pantalla. Y no es este un asunto trivial. Parece mentira que una zona tan iluminada, resulte a la vez tan oscura.

El Spectrum dispone de una pantalla de alta resolución de formato único, que funciona según el método llamado «mapeado en bits» *(bit-mapped).* Decimos «de formato único» porque es muy común en otros ordenadores el disponer de dos o más posibles organizaciones —que se caracterizan por dar más colores cuanta menor sea la resolución.

La complejidad de la distribución de la pantalla aumenta enormemente en cuanto nos salimos del BASIC. Para proceder por orden, empezaremos a este nivel.

Es posible referirnos a la pantalla de dos modos, aunque más tarde veremos que es uno solo. Podemos hablar de casillas y de *«pixels».*

Convendremos en llamar **casillas de caracteres** a los espacios dedicados a caracteres. Si escribe:

PRINT AT 12,16;INVERSE 1;" "

se dibujará en mitad de su pantalla un cuadro de color. Esta superficie es la que llamamos casilla. La resolución —el número de casillas del que se dispone— es de 32 columnas por 24 líneas, siendo las dos últimas no accesibles directamente.

Cada casilla tiene asociado un *byte* en la llamada área de atributos. Este *byte* guarda información sobre el color de la «tinta», el «papel» y la condición de parpadeo o brillo. Más adelante veremos más sobre ello.

El término **pixel** *(picture element,* elemento de imagen) hace referencia al punto más pequeño que puede usarse en la pantalla del Spectrum. Haciendo

PLOT 128,88

dibujaremos un *pixel* en mitad de la pantalla. Es posible dibujar 256 *pixels* en horizontal y 176 en vertical.

Como se indica de manera implícita en los ejemplos anteriores, nosotros accedemos a un punto a través de sus coordenadas. En el caso de las casillas, en primer lugar se indicará la coordenada vertical y, seguida de una coma, la coordenada horizontal. El origen de coordenadas —el punto 0,0— es la esquina superior izquierda. En lo que a *pixels* se refiere, la primera coordenada es la horizontal, y la segunda la vertical. El origen está en la esquina inferior izquierda. Como vemos, en este último caso, la semejanza es total con las coordenadas cartesianas.

Muchas veces es necesario cambiar la posición del origen y/o ampliar o reducir los dibujos. Las transformaciones a realizar son sencillísimas. Veamos el programa 1 (fig. 1), que incluye todas las transformaciones a realizar para el dibujo de cualquier función, que se indicará en la línea 160.

Debemos tener en cuenta que los *pixels* y las casillas no son sino diferentes maneras de ver una mis-

ma cosa. Un caracter cualesquiera está formado a base de puntitos oscuros o iluminados. Estos puntitos no son otra cosa que *pixels*. Así, una casilla es una matriz 8 × 8 puntos.

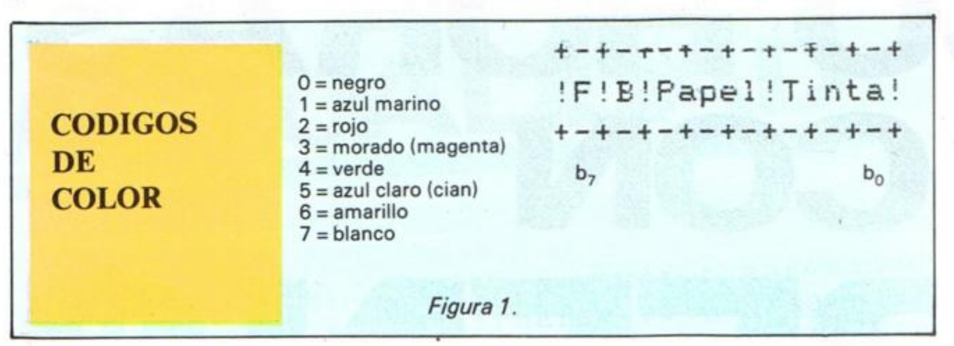


Figura 1.

Atributos

Consideremos ahora el abandonado tema de los atributos. Como mencionábamos, para cada casilla, existe un *byte* de atributos. El valor de este octeto puede expresarse como

1)+8*Código papel+Código tinta

Este valor puede obtenerse mediante el comando ATTR (x,y).

Dejemos de lado el parpadeo o el brillo, mantengamos únicamente la información de los colores.

Cuando ponemos cualquiera de los *bits* del mapa que forma la pantalla, a uno, inmediatamente dicho punto tomará el color de la tinta que hallamos asignado. Los *bits* que estén a cero, adoptarán el color del papel. Veamos un ejemplo en el que se escribe A's en toda la pantalla (programa 2). La línea 30 tiene por misión el poner todos los atributos de la pantalla a un valor determinado.

¿Cómo es posible?

Hemos visto, aunque de manera indirecta, que los datos introducidos en ciertos lugares de la memoria RAM se reflejan en la pantalla de alguna extraña manera. En efecto, entre las direcciones 16384 (4000H) y 22527 (57FFH) está almacenada la información sobre *pixels*. Todo punto —caracterizado por sus coordenadas— tiene asociado un *bit* que, como siempre, podrá estar a cero o a uno (dando, por tanto, el color del papel o de la tinta). Existe otra segunda área, la ya comentada de atributos, que asocia un *byte* a cada casilla. La ordenación es muy simple. La primera posición (22528, 5800H) corresponde a la esquina superior izquierda. El siguiente dato será el correspondiente a la casilla situada a la derecha de la anterior. Y seguimos en la misma línea hasta la posición 32. La siguiente ya corresponde al segundo renglón. Así, hasta completar 32*24 = 768 *bytes* (300H).

Otro punto de interés es la manera en la que los datos almacenados en memoria pueden salir a la pantalla. En detalle el procedimiento es complejo, pero podría resumirse diciendo que en la ULA —circuito integrado especialmente diseñado para el Spectrum— existen una serie de circuitos que continuamente interrumpen el acceso a memoria del microprocesador, y obtienen del área de pantalla la información adecuada. Estos datos debidamente procesados se mandan al circuito modulador, que los envía a la toma de antena en una portadora de alta frecuencia.

La distribución del área de pantalla

Volvemos a los *pixels*. Los puntos correspondientes a una misma «rebanada» horizontal de una misma casilla están almacenados en un *byte*. Esto supone grandes ventajas de orden práctico, ya que nos permite simplificar enormemente la tarea de escritura de caracteres en pantalla.

Los aficionados a los juegos se habrán preguntado muchas veces por qué al cargar un dibujo de presentación, las líneas van apareciendo en tan extraño orden. Un efecto semejante es el que muestra el programa 3. Al igual que en el caso anterior, mientras que la me-

MAPA DE LA DISTRIBUCION DE LA PANTALLA

Trazo fino: cambio de línea.
Trazo grueso: cambio de bloque.

4000	4001	4002	4003	4004	4005	. . . .	401D	401E	401F
4100	4101	4102	4103	4104	4105		411D	411E	411F
4200	4201	4202	4203	4204	4205	. . . .	421D	421E	421F
4300	4301	4302	4303	4304	4305		431D	431E	431F
4400	4401	4402	4403	4404	4405	. . . .	441D	441E	441F
4500	4501	4502	4503	4504	4505		451D	451E	451F
4600	4601	4602	4603	4604	4605	. . . .	461D	461E	461F
4700	4701	4702	4703	4704	4705		471D	471E	471F
4020	4021	4022	4023	4024	4025	. . . .	403D	403E	403F
4120	4121	4122	4123	4124	4125	. . . .	413D	413E	413F
4720	4721	4722	4723	4724	4725	. . . .	473D	483E	463F
4040	4041	4042	4043	4044	4045	. . . .	405D	405E	405F
4740	4741	4742	4743	4744	4745	. . . .	475D	475E	475F
4060	4061	4062	4063	4064	4065	. . . .	407D	407E	407F
4760	4761	4762	4763	4764	4765	. . . .	477D	477E	477F
4080	4081	4082	4083	4004	4085	. . . .	409D	409E	409F
4780	4781	4782	4783	4784	4785	. . . .	479D	489E	469F
40A0	40A1	40A2	40A3	40A4	40A5	. . . .	40BD	40BE	40BF
47A0	47A1	47A2	47A3	47A4	47A5	. . . .	47BD	47BE	47BF
40C0	40C1	40C2	40C3	40C4	40C5	. . . .	40DD	40DE	40DF
47C0	47C1	47C2	47C3	47C4	47C5	. . . .	47DD	47DE	47DF
40E0	40E1	40E2	40E3	40E4	40E5	. . . .	40FD	40FE	40FF
47E0	47E1	47E2	47E3	47E4	47E5		47FD	47FE	47FF
4800	4801	4802	4803	4804	4805		481D	481E	481F
4F00	4F01	4F02	4F03	4F04	4F05	. . . .	4F1D	4F1E	4F1F
4820	4621	4822	4823	4824	4825	. . . .	483D	483E	463F
. . .	. . .	. . .	. . .	. . .	. . .		. . .	. . .	. . .
4FC0	4FC1	4FC2	4FC3	4FC4	4FC5	. . . .	4FDD	4FDE	4FDF
48E0	48E1	48E2	48E3	48E4	48E5	. . . .	48FD	48FE	48FF
4FE0	4FE1	4FE2	4FE3	4FE4	4FE5		4FFD	4FFE	4FFF
5000	5001	5002	5003	5004	5005		501D	501E	501F
5700	5701	5702	5703	5704	5705	. . . .	571D	571E	571F
5020	5021	5022	5023	5024	5025	. . . .	503D	503E	503F
. . .	. . .	. . .	. . .	. . .	. . .		. . .	. . .	. . .
57C0	57C1	57C2	57C3	57C4	57C5	. . . .	57DD	57DE	57DF
50E0	50E1	50E2	50E3	50E4	50E5	. . . .	50FD	50FE	50FF
57E0	57E1	57E2	57E3	57E4	57E5	. . . .	57FD	57FE	57FF

```
  10  REM Programa 1
  20  REM
  30  LET X0=256/2: LET Y0=176/2
     : REM Posicion del origen d
     e coordenadas
  40  LET XF=20: LET YF=20: REM
     Factores de escala. Mayor q
     ue uno, agrandan, menos, di
     sminuyen.
  50  PLOT 10,Y0: DRAW 236,0: PL
     OT X0,10: DRAW 0,156: REM D
     ibujo de los ejes.
  55  REM Dibujo de escalas
  60  LET UX=XF: LET UY=YF
  70  IF UX>100 THEN  LET UX=UX/
     10: GO TO 70
  80  IF UY>80 THEN  LET UY=UY/1
     0: GO TO 80
  90  IF UX<5 THEN  LET UX=10*UX
     : GO TO 90
 100  IF UY<5 THEN  LET UY=10*UY
     : GO TO 100
 110  LET DX=X0-UX*INT (X0/UX):
     LET DX=DX-UX*(DX>UX/2): FOR
      I=1 TO 236/UX: PLOT UX*I+D
     X,Y0-2: DRAW 0,5: NEXT I
 120  LET DY=Y0-UY*INT (Y0/UY):
     LET DY=DY-UY*(DY>UY/2): FOR
      I=1 TO 156/UY: PLOT X0-2,U
     Y*I+DY: DRAW 5,0: NEXT I
 130  PRINT AT 0,0;"Intervalos X
     =";XF/UX,"Y=";YF/UY
 140  REM Ejemplo
 150  FOR I=5 TO 250: LET X=(I-X
     0)/XF
 160  LET Y=TAN X
 170  LET YC=YF*Y+Y0: IF YC>0 AN
     D YC<176 THEN   PLOT I,YF*Y
     +Y0
 180  NEXT I
```

```
  10  REM Programa 2
  20  REM
  30  INK 7: PAPER 0: BORDER 0:
     CLS
  40  FOR I=0 TO 7
  50  READ A
  60  POKE 256*I+16384,A
  70  NEXT I
 100  DATA BIN 00000000
 110  DATA BIN 00111100
 120  DATA BIN 01000010
 130  DATA BIN 01000010
 140  DATA BIN 01111110
 150  DATA BIN 01000010
 160  DATA BIN 01000010
 170  DATA BIN 00000000
 180
 190
 200  REM Programa 3
 210  REM
 220  CLS
 230  FOR I=16384 TO 22527
 240  POKE I,255
 250  NEXT I
 260  PAUSE 0
```

```
;           PROGRAMA 4

; PRINT AT 10,12;"*";TAB 9;"-"
; ------------------------------

;Abrimos el canal 2, la pantalla

          LD   A,2
          CALL 1601H

          LD   DE,TEXTO
BUCLE     LD   A,(DE)
          BIT  7,A; Fin de texto
          RET  NZ
          RST  10H
          INC  DE
          JR   BUCLE

TEXTO     DEFB AT,10,12,'+'
          DEFB TAB,9,0,'-'
          DEFB ENTER
          DEFB 80H

ENTER     EQU  13
AT        EQU  22
TAB       EQU  23
```

moria se altera de forma consecutiva, el efecto se muestra de manera alterna.

A primera vista podemos distinguir tres bloques idénticos. Los tres tienen una ordenación igual, por lo que estudiaremos únicamente el primero, que va de la dirección 4000H a la 47FFH. Un análisis más detallado nos lleva a descubrir que en cada bloque hay 8 bandas. Tres bloques multiplicado por 8 bandas por bloque, nos dan un total de 24 bandas... ¡Estas bandas son las líneas de texto! En efecto, estas líneas tienen una altura de «8 pisos». A lo ancho, cada una ocupa 256/8 = 32 *bytes*. 32 *bytes*/línea × 8 líneas por banda × 8 bandas = 2048, los 2 K que se obtienen restando direcciones de principio y final.

Generalmente, nuestro mayor interés reside en escribir por líneas. Esto es sencillo, pues partiendo de la dirección del primer ladrillo de la casilla, las siguientes se obtienen sumando 256 a las anteriores, hasta conseguir los ocho pisos. Las primeras filas de casilla se caracterizan porque tienen los 3 *bits* menos significativos del *byte* alto a cero.

El cambio de bloque supone nuevas complicaciones, que se resolverán en cada caso de forma diferente. (Obsérvese, por ejemplo, las soluciones adoptadas en los programas de *scroll*).

La figura 2 muestra para cada las direcciones asociadas a cada rebanada.

PLOT a velocidad de luz

Trabajando en lenguaje ensamblador —lo que se conoce incorrectamente como código máquina— muchas veces deseamos conocer a partir de las coordenadas de un punto la dirección en la que

```

;      ************************
;      ***  SCROLL VERTICAL  ***
;      ************************

;(c) Luis Miguel BRUGAROLAS 1985


           EXX
           LD   BC,0708H
CONT       EXX

           LD   B,00
           LD   HL,4100H

BLOCK      LD   E,L
           LD   D,H
           DEC  D
           JR   TRANSF

LINE       EXX
           LD   L,A
           JR   NC,TRANSF
           INC  H
TRANSF     LD   C,32
           LDIR
           LD   A,-32
           ADD  A,E
           LD   E,A
           JR   NC,JMP
           INC  D
JMP        EXX
           DJNZ LINE
           LD   B,07
           EXX

           LD   A,L
           OR   A
           JR   Z,NW_BLK
NO_FIN     LD   A,-7
           ADD  A,H
           LD   H,A
NXTBLK     LD   C,32
           LDIR
           LD   A,-32
           ADD  A,L
           LD   L,A
           JR   NC,BLOCK
           INC  H
           JR   BLOCK

NW_BLK     LD   A,H
           CP   58H
           JR   NZ,NXTBLK

           XOR  A
           LD   HL,57E0H
           LD   (HL),A
           LD   DE,57E1H
           LD   C,31
           LDIR

           EXX
           DEC  C
           JR   NZ,CONT
           EXX

           LD   DE,5800H
           LD   HL,5820H
           LD   BC,02E0H
           LDIR
           RET
```

se encuentra el punto a dibujar, y cuál de los *bits* es. Aunque parezca mucho pedir, existe en ROM una subrutina que nos viene al dedillo. Si cargamos en B la coordenada 'Y', en C la 'X', y llamamos a la subrutina de la dirección 22AAH, volveremos guardando en HL la dirección del *byte* a alterar. En el registro A tendremos el orden de nuestro *bit.* Pongamos el ejemplo concreto de hacer un PLOT en el punto 200,100. Escribiremos el programa de la figura 3.

```
01C864   01       LD     BC, 256*100+200; X(C)=200, Y(B)=100
CDAA22   02       CALL   22AAH
47       03       LD     B, A
04       04       INC    B
3E01     05       LD     A, 01
OF       06  ROT  RRCA
1OFD     07       DJNZ   ROT
B6       08       OR     (HL)
77       09       LD     (HL), A
C9       10       RET
```

Figura 3.

La simulación de un OVER 1 se llevará a cabo mediante la función XOR. Cambiaremos la línea 3 por un LD A,OFEH, y la 6 por el consabido XOR (HL).

Escritura de textos en programa ensamblador

Vamos a pasar a otro asunto más interesante si cabe. Se trata de escribir textos en pantalla o impresora desde programas en código máquina (ahora sí).

Las manipulaciones que requiere la operación son muy sencillas. Cuando vayamos a escribir algo por primera vez, debemos asegurarnos de que el canal de pantalla se encuentre abierto. Y la mejor forma de hacerlo, es forzando la situación. Una vez realizado esto, simplemente cargamos en A el código del caracter a imprimir, y hacemos un RST 10H. El caracter aparecerá en pantalla. Con objeto de controlar su posición, podemos emplear los códigos de control comprendidos entre 16 y 23. Así, si queremos hacer el equivalente de PRINT AT 10,12;"*";TAB 13;"–", escribiremos un programa como el 4 (figura 5). Para el PRINT AT, mandamos el código AT, y a continuación las dos coordenadas del lugar donde deseamos escribir y el caracter o caracteres a imprimir. El PRINT TAB guarda una ligera sorpresa, en el sentido que requiere 2 *bytes*, como siempre el primero es el menos significativo. Es curioso, pero el más significativo no lo es en absoluto, pues se ignora por completo.

Esta no es la única manera de escribir texto en pantalla aunque sí la más útil. Indicaremos nada más que la mencionada subrutina no perturba el contenido de los registros —sí del juego alterno o imagen—, por lo que puede usarse directamente.

```
10  REM Programa 8
20  REM
30  DATA 33,0,72,17,0,118,1,0,
   8,237,176,201
40  CLEAR 30000
50  LET ADDR=23296
60  FOR I=0 TO 11: READ A: POK
   E I+ADDR,A: NEXT I
70  REM Almacenamos en Memoria
80  FOR I=0 TO 14: CLS
90  PLOT 20,80: DRAW 200,0,(I-
   7)/6
100  POKE ADDR+5,256-(I+1)*8
110  RANDOMIZE USR ADDR
120  NEXT I
130  REM Visualizacion
140  POKE ADDR+5,72
150  FOR I=0 TO 14: PAUSE 3: PO
    KE ADDR+2,256-(I+1)*8: RAND
    OMIZE USR ADDR: NEXT I
160  FOR I=14 TO 0 STEP -1: PAU
    SE 3: POKE ADDR+2,256-(I+1)
    *8: RANDOMIZE USR ADDR: NEX
    T I
170  GO TO 150
```

```
;*************************
;** PROGRAMAS DE SCROLL **
;**    EN HORIZONTAL    **
;*************************
;(c) Luis Miguel BRUGAROLAS 1985
;  SCROLL A LA DERECHA
;-----------------------
        LD   HL,4000H
        LD   C,24*8
LINER   LD   B,32
        XOR  A
ROTR    RR   (HL)
        INC  HL
        DJNZ ROTR
        DEC  C
        JR   NZ,LINER
        RET
;SCROLL UNA LINEA A LA DERECHA
;-----------------------------
        LD   HL,5000H
        LD   C,08
LINE1R  LD   B,32
        XOR  A
ROT1R   RR   (HL)
        INC  L
        DJNZ ROT1R
        LD   A,-32
        ADD  A,L
        LD   L,A
        INC  H
        DEC  C
        JR   NZ,LINE1R
        RET
;  SCROLL A LA IZQUIERDA
;-------------------------
        LD   HL,57FFH
        LD   C,24*8
LINEL   LD   B,32 .
        XOR  A
ROTL    RL   (HL)
        DEC  HL
        DJNZ ROTL
        DEC  C
        JR   NZ,LINEL
        RET
;SCROLL DE UNA LINEA A LA IZQ
;-----------------------------
        LD   HL,501FH
        LD   C,08
LINE1L  LD   B,32
        XOR  A
ROT1L   RL   (HL)
        DEC  L
        DJNZ ROT1L
        LD   A,32
        ADD  A,L
        LD   L,A
        INC  H
        DEC  C
        JR   NZ,LINE1L
        RET
```

Cambios de juego de caracteres

La información sobre el «perfil» de los caracteres se halla almacenada en ROM, entre las direcciones 3D00 y 3FFFH. La selección se realiza mediante direccionamiento indirecto al contenido de ¡cómo no!, una de las variables del sistema. En CHARS (23606 y 7) se almacena la dirección de comienzo de la tabla menos 256. Esto es, el *byte* bajo queda inalterado, y el alto queda disminuido en una unidad. Si creamos otra, esta vez en RAM, y modificamos consecuentemente este puntero, podemos disfrutar de caracteres a medida. Incluso podemos usarlo como medida de protección en nuestros programas.

Programas de *Scroll* suavecito

Aunque el término **«Scroll»** no es excesivamente español, se halla asumido perfectamente por los usuarios del Spectrum. Por si todavía hay algún despistadillo, diremos que se trata de un desplazamiento de la imagen en pantalla. El *Scroll* que dispone el ordenador tiene la ventaja de ser muy rápido, pero es hacer un USR 3582). Con objeto de proporcionar al Spectrum un aspecto más profesional, se ha escrito una rutina de *scroll* vertical muy suave. Merece la pena probarse.

Igualmente, hemos escrito programas de *Scroll* en horizontal, tanto para la pantalla entera, como para una única línea. Este último puede resultar de gran interés para juegos del tipo de una autopista que se ve por encima.

Cambio de pantallas

En muchas ocasiones resulta muy interesante mover rápidamente complicadas figuras. Como no es posible dibujarlas en tan poco tiempo, debemos buscar un método alternativo. Este método existe. Efectivamente, es posible transferir bloques enteros de pantalla de un lugar a otro de la memoria, dando ilusión de movimiento. El programa 8 ilustra el efecto. Aquí no se ha realizado la transferencia de la pantalla entera, sino sólo del tercio central. De esta manera, tenemos la oportunidad de guardar más imágenes, y por tanto de conseguir un mayor realismo. En cualquier caso, siempre podemos guardar la pantalla completa, incluso con el área de atributos.

Luis M. Brugarolas

```
100 DATA 217,1,8,7,217,6,0,33,
    0,65,93,84,21,24,5,217
110 DATA 111,48,1,36,14,32,237
    ,176,62,224,131,95,48,1,20,
    217
120 DATA 16,237,6,7,217,125,18
    3,40,17,62,249,132,103,14,3
    2,237
130 DATA 176,62,224,133,111,48
    ,211,36,24,208,124,254,88,3
    2,238,175
140 DATA 33,224,87,119,17,225,
    87,14,31,237,176,217,13,32,
    181,217
150 DATA 17,0,88,33,32,88,1,22
    4,2,237,176,201,33,0,64,14
160 DATA 192,6,32,175,203,30,3
    5,16,251,13,32,245,201,33,0
    ,80
170 DATA 14,8,6,32,175,203,30,
    44,16,251,62,224,133,111,36
    ,13
180 DATA 32,240,201,33,255,87,
    14,192,6,32,175,203,22,43,1
    6,251
190 DATA 13,32,245,201,33,31,8
    0,14,8,6,32,175,203,22,45,1
    6
200 DATA 251,62,32,133,111,36,
    13,32,240,201
500 LET SUM=0: LET DIR=23296
510 FOR I=0 TO 169
520 READ A: POKE DIR+I,A: LET
    SUM=SUM+A
530 NEXT I
540 IF SUM<>16363 THEN  BEEP .
    5,40: PRINT "ERROR EN EL DA
    TA": STOP
600 REM
610 REM DEMOSTRACION
620 REM
630 CLS : FOR i=1 TO 32*21: PR
    INT CHR$ (32+100*RND );: NE
    XT i
640 REM  Estas rutinas son rel
    o-calizables, pueden situar
    se en cualquier lugar de la
     memoria.
650 REM Scroll Vertical
660 RANDOMIZE USR dir: BEEP .1
    ,10: PAUSE 50
670 REM Scroll a la derecha
680 RANDOMIZE USR (dir+92): BE
    EP .1,30: PAUSE 50
690 FOR i=0 TO 7: RANDOMIZE US
    R (dir+92): NEXT i: BEEP .1
    ,10: PAUSE 50
700 REM Scroll a la izquierda
710 RANDOMIZE USR (dir+131): B
    EEP .1,30: PAUSE 50
720 FOR i=0 TO 7: RANDOMIZE US
    R (dir+131): NEXT i: BEEP .
    1,10: PAUSE 50
730 INPUT "Fila (0-23):";F: IF
     F<0 OR F>23 THEN  GO TO 73
    0
740 LET H=8*INT (F/8)+64
750 LET L=32*(F-8*INT (F/8))
760 POKE dir+110,l: POKE dir+1
    11,h: POKE dir+149,l+31: PO
    KE dir+150,h
770 REM Scroll de lineas
780 REM A la derecha
790 RANDOMIZE USR (dir+109): B
    EEP .1,30: PAUSE 50
800 FOR i=0 TO 7: RANDOMIZE US
    R (dir+109): NEXT i: BEEP
    1,10: PAUSE 50
```



Hardware

Al mirar cualquier circuito electrónico entramos en un mundo misterioso habitado por los seres más extraños que pudiésemos imaginar. Allí conviven raras especies de ciempiés negros y planos, con multitud de patas brillantes, conocidos como *«chips»,* transistores de varias especies, grandes y alargados con placas metálicas, o como pequeñas cabezas, metálicas o plásticas, y dos o tres patas de alambre, resistencias como gusanos cilíndricos con el cuerpo pintado con anillos de colores, condensadores con forma de pequeños barriles metálicos, o pequeñas gotas de plástico que se hubiesen solidificado repentinamente, o bien en forma de lenteja. Esta fauna exótica vive sobre un curioso terreno, llamado «placa de circuito impreso», o «placa» a secas, alimentándose de electricidad y realizando funciones más o menos desconocidas. A estos bichos y a sus costumbres dedicaremos este artículo.

El principio: voltios, amperios...

Bueno, pues todos estos bichos, desde la más pequeña resistencia hasta un ordenador, dependen y viven de la electricidad, que es esa cosa que se esconde en los enchufes de las paredes y que suele dar calambres. Pero, ¿qué es la electricidad? Pueden decirse muchas cosas sobre ella, pero iremos a lo práctico. En principio, es algo que tiene que ver con unas fuerzas que se producen entre partículas «cargadas», y que nos interesa que se muevan (y eso es todo, porque no es cosa de dar un curso de Física).

Podemos ver dos cosas interesantes en la electricidad: una, la que hace que esas partículas cargadas se muevan (a las que, a partir de ahora, llamaremos electrones) y otra, que es el movimiento. Pensemos en un ejemplo típico: una pequeña cascada que cae sobre un molino de agua. La rueda se mueve al caer el agua en las paletas (figura 1). Y se mueve con más rapidez si la catarata es más alta, aunque no caiga más agua, pero también con la misma altura y más agua giraría con mayor rapidez. Es decir, hay dos aspectos que nos interesan, la altura y la cantidad de agua. Con la electricidad ocurre lo mismo, nos interesan la fuerza, llamada «tensión», que se mide en voltios (V), y la cantidad, llamada «corriente» y medida en amperios (A).



Nuestra labor de montaje del circuito a punto de terminar. Todos los componentes e instrumental necesario se hallan distribuidos alrededor de la placa que soporta el circuito.

De todas formas, hay que decir algo más. Ambas pueden tener un signo (+) o (–). Normalmente hay un punto en el circuito llamado «masa» o «tierra», que se representa por ⏚ o ⏚ y con respecto a la cual se miden las tensiones. Si la tensión es mayor que la masa, es positiva, y si es menor, negativa. Es como la altura de las montañas, se toma toda con respecto al nivel del mar, y podemos comparar alturas porque tenemos un punto «cero» con respecto al que podemos medir. El signo de la corriente se toma de otra forma. En general va siempre de las tensiones positivas mayores a las menores; es decir, siempre de positivas a negativas o de positivas a masa o de masa a negativas. Pensemos de nuevo en el agua. Si llueve en la cima de una montaña, o en donde sea, el agua va siempre buscando los puntos más bajos, de menor altura. Así se comporta la corriente.

Potencia

¿Qué es mejor: 100 voltios y 1 amperio o 1 voltio y 100 amperios? La respuesta es que depende. Pero además podemos hacer comparaciones gracias a la potencia, que es el producto de tensión por corriente y el resultado se mide en vatios (W). En el cuadro 1 aparecen las fórmulas que les relacionan. Se ve que en el ejemplo la potencia es la misma en los dos casos. De todas formas, la potencia puede ser creada o consumida. Después volveremos sobre esto.

Un último detalle sobre las unidades es que a veces son demasiado grandes o demasiado pequeñas. Por eso se suelen emplear prefijos para simplificar la escritura. Por ejemplo, en lugar de decir «un millón de vatios», decimos «un megavatio» y escribimos 1 MW. Ade-

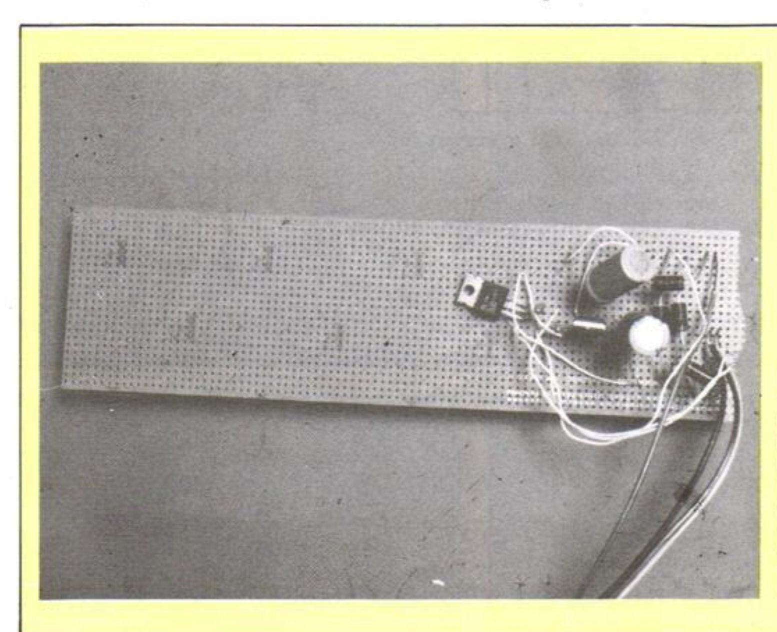
más, en electrónica, trabajaremos más bien con milésimas de amperio y de vatio, que son los miliamperios (se escribe «mA») y milivatios (mW). De todas formas, los prefijos más usados son los indicados en el cuadro 2.

Es curioso que las abreviaturas de los grandes son letras mayúsculas, mientras que las de los pequeños son minúsculas, aunque no es una casualidad, sino para recordarlos mejor.

en general, los metales. La madera, el cristal o los plásticos son buenos aislantes.

En cuanto a lo del grosor o de la longitud, pensemos en una corriente de agua. Cuanto más estrecha sea la manguera, peor pasará el agua. Y, desde luego, si al agua le cuesta pasar, habrá que hacer tanto más fuerza cuanto más larga sea la manguera.

A esto se le llama resistencia; y lo que indica es cómo le es de difícil a la electricidad atravesar un material. Para medir esa resistencia, podemos hacer un experimento. Cogemos una barra de hierro, le aplicamos una tensión de 3 voltios y medimos la corriente. Supongamos que sale 1 amperio. Si le aplicamos ahora 15 voltios, obtenemos una corriente de 5 amperios. Si fuesen 300 voltios, saldrían 100 amperios. Dividiendo la tensión por la corriente se obtiene siempre el mismo número, el 3, que es el valor de la resistencia. Eso, bien dicho, es la ley de Ohm, que dice que «la resistencia es igual a la tensión partida por la corriente» (en el cuadro 3a aparecen las relaciones), y la unidad de resistencia son los ohmios (se prepresentan por Ω).



La misma placa uniprint pero vista por el lado de los componentes, donde se puede ver que las distintas pistas están unidas con cables para unir los componentes.

Resistencia

Hemos dicho que la corriente eléctrica son partículas cargadas que se mueven. Pero no puede ir igual de cómoda por todos los terrenos. Los hay más y menos agradables. A los agradables, que son aquellos en los que, con una tensión pequeña conseguimos corrientes más bien grandes, se les llama «conductores». A los que, por mucha tensión que les apliquemos no conseguiremos corriente, se les llama «aislantes». En que un material sea aislante o conductor influyen cosas como lo largo que sea, la superficie o la naturaleza del material. Así, la plata o el cobre son buenos conductores y,

Resistencias «de verdad»

¿Qué ocurre cuando necesitamos resistencias? ¿Dónde y cómo se compran? ¿Son caras? Estas son algunas de las preguntas que aparecen cuando hay que trabajar con resistencias y que contestaremos ahora.

Lo primero que hemos de saber es que no se fabrican todas las resistencias posibles; sería carísimo de fabricar los 10.000.000 de resistencias que van desde las de 1 ohmio hasta las de 10 Megaohmios (10 «Megas» para los amigos), y siempre habría alguien que necesitase una de 97.321,438 ohmios, que no estaría incluida en la lista, así que ésta no serviría para nada.

Por otra parte, si queremos una resistencia de 1.500 ohmios podemos fabricarla para que salga exactamente de 1.500, pero es tremendamente cara. Por el contrario, es muy barato fabricarlas de 1.500 ohmios, más o menos. El «más o menos» quiere decir que puede variar un poco, por encima o por debajo, del valor que queramos, pero siempre sin pasar de determinados valores. Así, nos venden resistencias con «tolerancias» del 5 o del 10 por ciento.

Eso quiere decir que el valor de una resistencia de 1.500 ohmios y 5 por ciento de tolerancia variará entre 1.425 y 1.575 ohmios, pero no 1.350 ó 1.700 ohmios. Además, en la mayoría de los casos, los valores de la resistencia estarán más cerca de los 1.500 y del 10 por ciento, el valor estaría entre 1.350 y 1.650 ohmios.

Así pues, se escogen los valores de las resistencias de forma que, cada valor de resistencia más sus límites de tolerancia nos den todos los valores posibles por «década». Así, la serie E12 (o del 10 por ciento) llamada así por tener 12 resistencias, comprende estos valores:
1, 1,2, 1,5, 1,8, 2,2, 2,7, 3,3, 3,9, 4,7, 5,6, 6,8
que pueden ir multiplicados por

10,100, 1.000, 10.000, 100.000 ó un millón. Veamos que, aproximadamente «recubren» todos los valores posibles. Por ejemplo, 3,3 más el 10 por ciento de 3,3 es 3,63; y 3,9 menos el 10 por ciento es 3,51; es decir, se recubren. Así ocurre con los demás valores, aunque en algún caso (pocos) no se cumplan.

Se fabrican series con tolerancia del 20 por ciento (6 valores por década o E6), 10 por ciento (vista antes), 5 por ciento, 2 por ciento y 1 por ciento. La más usada es la del 5 por ciento, cuyos valores son: 1, 1,1, 1,2, 1,3, 1,5, 1,6, 1,8, 2, 2,2, 2,4, 2,7, 3, 3,3, 3,6, 3,9, 4,3, 4,7, 5,1, 5,6, 6,2, 6,8, 7,5, 8,2, 9,1 (24 valores).

El problema suele ser que, en muchas tiendas, los valores de resistencias que tienen con tolerancias del 5 por ciento son los valores del 10 por ciento (los doce valores anteriores, no los 24).

Alguien puede preguntarse, ¿cómo pueden funcionar los montajes con esos «errores» del 5 o del 10 por ciento? Bueno, pues, afortunadamente, esos errores no suelen tener gran importancia. Además, los diseños se hacen teniendo en cuenta las tolerancias y, bueno, pues las cosas terminan marchando.

Números y colores

Ya sabemos cuáles son los valores posibles. Ahora bien, ¿cómo sabemos cuál es el valor de una resistencia? No se suelen utilizar números, no sea que al soldar la resistencia a la placa quede oculto el valor. Por eso se emplean bandas de colores, cada una con un significado. Así, las dos primeras indican el valor de la resistencia, la tercera es el multiplicador y la cuarta, la tolerancia. En la figura 2 aparecen los colores utilizados y un ejemplo.

Otro detalle curioso es la jerga del gremio. No se suele decir «cuatro mil setecientos ohmios», sino «cuatro K siete», ni «ocho millones doscientos mil ohmios», sino «8 mega 2». Es más cómodo y queda mejor.

Potencia en las resistencias

Al principio vimos qué era la potencia, y que tiene que ver con la energía, y el calor es una forma de energía. Lo que interesa ahora es cuánta energía, o cuánto calor, o cuánta potencia puede «disipar» o resistir, una resistencia. Así, si una resistencia soporta 100 voltios y 10 amperios eso quiere decir que debe disipar 1.000 vatios, vamos, una buena estufa.

Nosotros no nos moveremos normalmente con esos números. Lo habitual será manejar resistencias de cuarto de vatio, medio vatio, o, como mucho, uno o dos vatios, aunque las haya mucho mayores.

Para saber cómo de grande debe ser nuestra resistencia, nos basta con conocer la tensión o la corriente. Las fórmulas correspondientes aparecen en el cuadro 3.

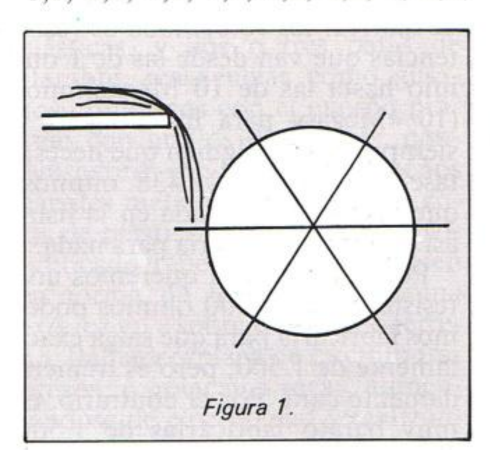


Figura 1.

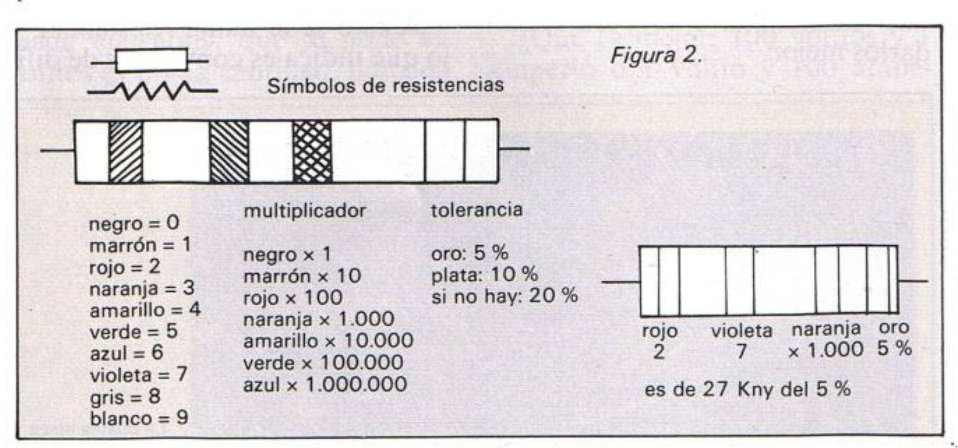
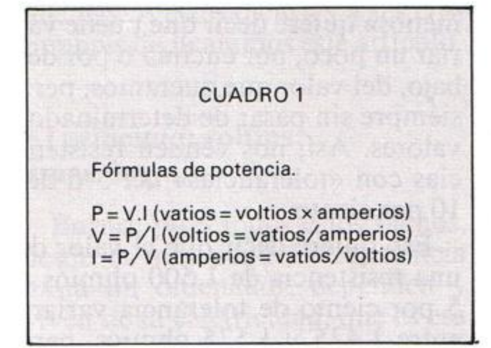


Figura 2.



CUADRO 1

Fórmulas de potencia.

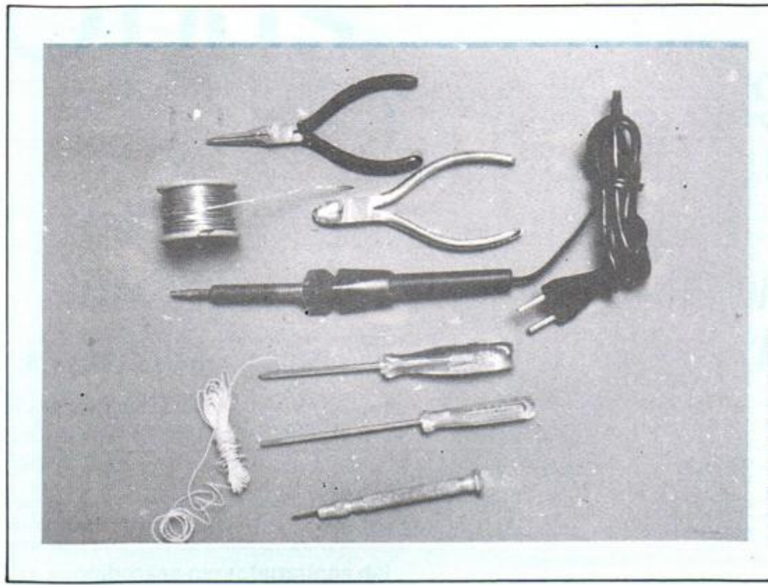
P = V.I (vatios = voltios × amperios)
V = P/I (voltios = vatios/amperios)
I = P/V (amperios = vatios/voltios)

CUADRO 2

Múltiplos y submúltiplos.
Mega (M) un millón = 1.000.000
Kilo (K) mil = 1.000
mili (m) milésima = 0,001
micro ( ) millonésima = 0,000.001
nano (n) milmillonésima = 0,000.000.001
pico (p) billonésima = 0,000.000.000.001

CUADRO 3

V = R.I	(voltios = ohmios × amperios)
R = V/I	(ohmios = voltios/amperios)
I = V/R	(tensión = voltios/ohmios)
$P = I^2R$	(vatios = $(\text{amperios})^2$ × ohmios)
$P = V^2/R$	(vatios = $(\text{voltios})^2$/ohmios)



Soldador, estaño, destornilladores y alicates de corte y punta fina son necesarios para emprender la tarea del montaje.

globo. A medida que lo vamos inflando, es más difícil meter aire. Si el globo fuese muy fuerte, llegaría un momento en que no podríamos seguir hinchándolo. Algo parecido ocurre con el fenómeno de la capacidad, almacenamos energía eléctrica que luego podemos utilizar para algo. La pega es que podemos almacenar bastante poca.

Uno de los problemas de las capacidades es que aquí no tenemos una relación tan clara entre tensión y corriente como en las resistencias. Así, un condensador no deja pasar la corriente eléctrica «continua» (que es de la que he-

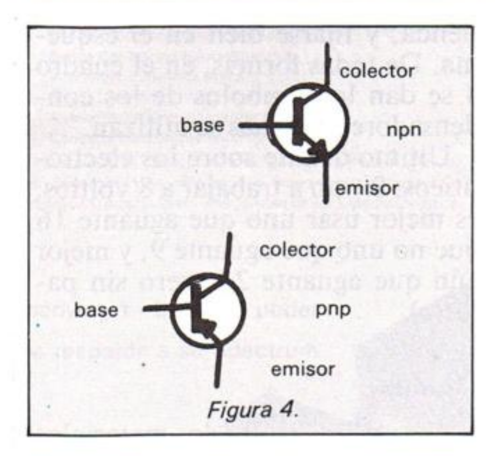


Figura 4.

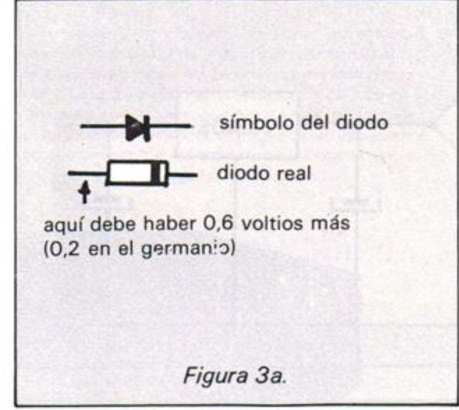


Figura 3a.

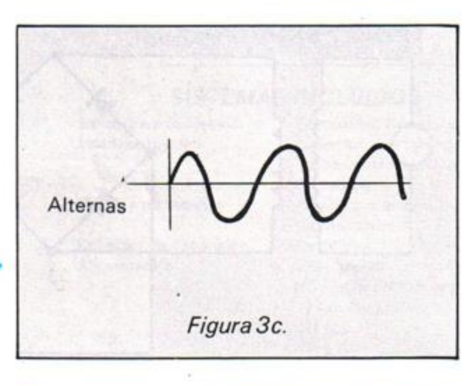
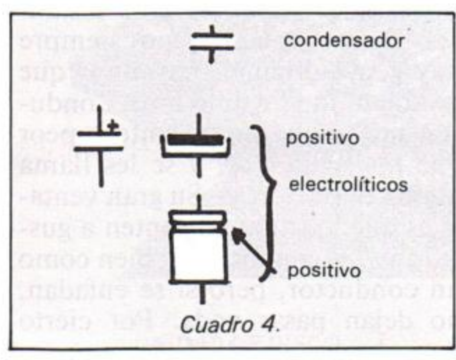


Figura 3c.



Cuadro 4.

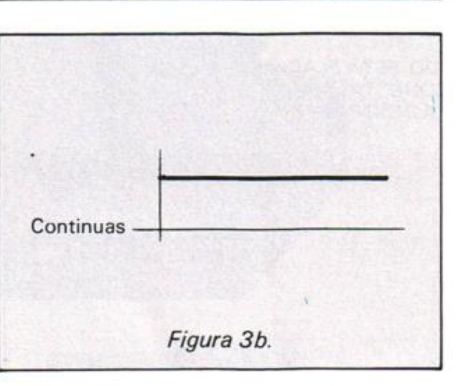


Figura 3b.

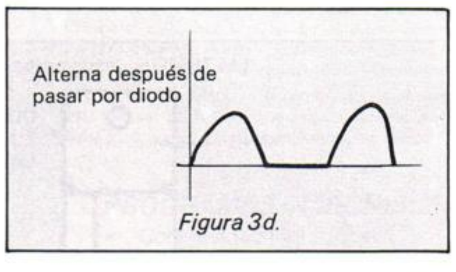


Figura 3d.

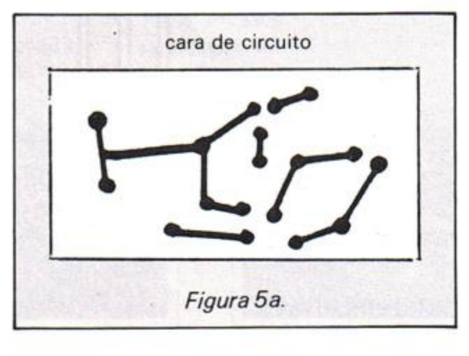


Figura 5a.

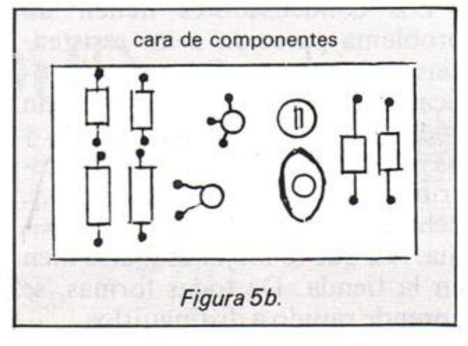


Figura 5b.

Resistencias ajustables

A veces, por ejemplo en los mandos de volumen, necesitamos poder variar el valor de una resistencia pero sin cambiar el componente. Para ello se emplean los potenciómetros o resistencias ajustables. Un potenciómetro tiene tres patas y un mango (en otros, más baratos, el mango se sustituye por una muesca en la que se encaja un destornillador, a éstos se les llama «potenciómetros de ajuste»).

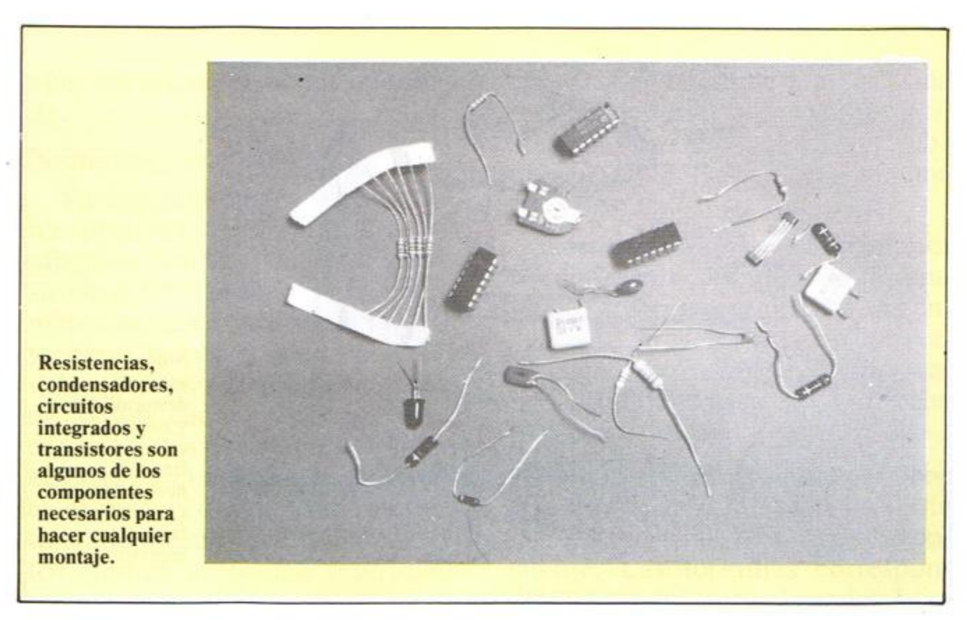
La idea es que hay una resistencia entre las dos patas de los extremos, pero no hace falta que la recorramos por completo, sino que podemos salir por la del centro. El valor de la resistencia se elige girando el mango y midiendo o bien observando el resultado del circuito. Los valores son los mismos que los de las resistencias de la E12, y no tiene sentido hablar de tolerancia. Las potencias de los normales suelen ser de 1/4 vatio o de 1/2 vatio y, en general, no dejan pasar demasiada corriente.

Capacidad

Para hacernos una idea de lo que es la capadidad, pensemos en un

mos hablado hasta ahora) y, en cuanto a la tensión, lo único que le importa es que no le apliquemos más tensión de la que puede aguantar, o reventaría como un globo demasiado hinchado.

La unidad de medida de capacidades es el Faradio, cuyo único defecto es que es enorme. Así que nadie pida un condensador de 600 ó 700 faradios, so pena de provocar muchas carcajadas. Normalmente los condensadores se miden en microfaradios, nanofaradios y picofaradios. Incluso se llega a tomar el picofaradio como unidad para los más pequeños, y así se habla de condensadores de 22 K, que no son 22.000 faradios, sino 22.000 picofaradios.



Resistencias, condensadores, circuitos integrados y transistores son algunos de los componentes necesarios para hacer cualquier montaje.

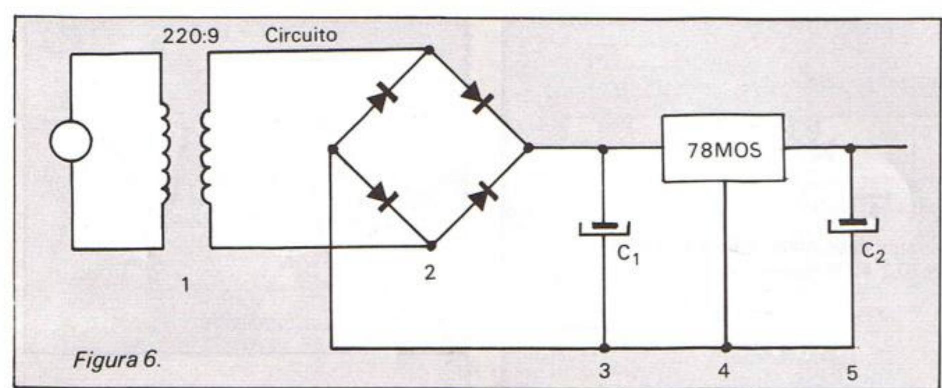


Figura 6.

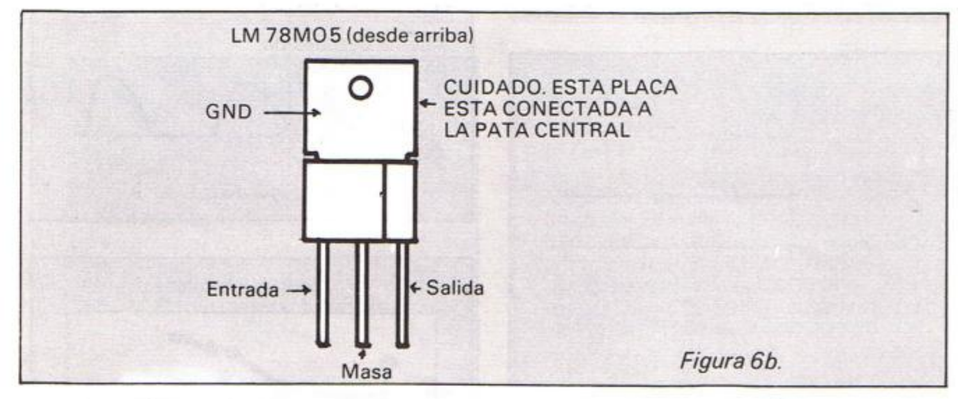


Figura 6b.

Condensadores

Los condensadores tienen un problema parecido a las resistencias en lo que se refiere a la identificación. Por desgracia no hay un código de colores, sino varios, y hay fabricantes que se limitan a escribir el valor de la capacidad y, debajo, el valor de la tensión máxima. Así que lo mejor es fijarse bien en la tienda. De todas formas, se aprende rápido a distinguirlos.

Hay unos condensadores muy particulares, que son los electrolíticos. Su ventaja es que son los que permiten mayores capacidades; su pega es que sólo trabajan si se sienten a gusto. Para ello, hay que enchufarlos de forma que uno de sus terminales, el positivo, esté a mayor tensión que el otro. De no ser así, pueden enfadarse mucho y llegan a estallar. De todas formas, los fabricantes sí dejan muy claro cuál es el positivo, poniendo un signo (+) cerca, o un (–) cerca del negativo, o un estrechamiento en el positivo. Si no se ve claro, es conveniente preguntarlo en la tienda, y fijarse bien en el esquema. De todas formas, en el cuadro 4 se dan los símbolos de los condensadores que más se utilizan.

Ultimo detalle sobre los electrolíticos. Si van a trabajar a 8 voltios, es mejor usar uno que aguante 16 que no uno que aguante 9, y mejor aún que aguante 25 (pero sin pasarse).

Diodos

Antes dijimos que los materiales pueden ser conductores o aislantes. Bueno, pues como siempre hay gente original, hay unos que no son ni lo uno ni lo otro. Conducen mejor que un aislante y peor que un conductor, y se les llama «semiconductores». Su gran ventaja es que, cuando se sienten a gusto, pueden conducir tan bien como un conductor, pero si se enfadan, no dejan pasar nada. Por cierto que el proceso de hacer que se sientan bien o mal se llama «polarización», y normalmente depende de cómo les apliquemos ciertas tensiones y de dónde las apliquemos.

Los materiales semiconductores más empleados son el germanio, que fue el primero que se empleó, y el silicio, que es el más usado hoy día (también en los *chips*), y el elemento semiconductor más sencillo es el «diodo».

Un diodo es algo que, según cuál sea su polarización, deja pasar la corriente o no. Así (ver la figura 3a), si tenemos un diodo de silicio, sólo conducirá si el terminal que

NO tiene la raya está a 0,6 voltios más que el que SI tiene la raya. De todas formas, viendo el símbolo del diodo, queda muy claro en qué sentido conduce (en el de la flecha). Esta tensión de 0,6 voltios es aproximadamente igual para todos los diodos de silicio y se le llama «tensión de conducción». Para los de germanio, es de 0,2 voltios.

Continua y alterna

Normalmente se han usado los diodos como «rectificadores». Veamos qué es eso. Hasta ahora hemos trabajado con tensiones y corrientes «continuas», es decir, si una tensión es de 5 voltios, siempre será de 5 voltios, y si una corriente es de 0,3 amperios, siempre será de 0,3 amperios. Pero ese no es el caso real. Si mirásemos cómo es la tensión de los enchufes de casa (o la corriente que podemos sacar de ellos) veríamos que su valor cambia constantemente (por eso se le llama alterna), de forma que unas veces es positiva y otras negativa (exactamente varía 50 veces por segundo), y esto se hace porque es más barato para transportarla. El problema es que la mayoría de los aparatos que utilizan dispositivos electrónicos necesitan tensiones «de alimentación» continuas, y aquí es donde entran los diodos. Si metemos una señal (corriente o tensión) alterna a un diodo, ésto *sólo* conducirá mientras que sea positiva, es decir, mientras que él esté «polarizado en directa», o sea, trabajando a gusto. Cuando deje de estar trabajando a gusto, se enfada, polariza en «inversa» y no deja pasar nada. De esta forma podemos conseguir una señal positiva o «rectificada» a partir de una alterna (figura 3b).

Transistores y circuitos integrados

Un transistor es un grifo de electricidad hecho con materiales semiconductores. Como necesita polarización para trabajar con agrado, podemos hacer que trabaje como queramos, y, lo que es mejor, puede amplificar señales pequeñas a grandes. Por eso se le llama «dispositivo activo». Como cada vez se pueden hacer más pequeños, pueden juntarse muchos, haciendo así «circuitos integrados», que hacen cosas más o menos complicadas, a bajo precio. Luego veremos un ejemplo.

Eso sí, hay que tener cuidado con los transistores. Son de varias clases, con diferentes misiones según el tipo. Hay unos, los p-n-p, que trabajan con tensiones negativas, otros, los n-p-n, trabajan con positivas. En cualquier caso, tienen normalmente tres terminales, emisor, base y colector. Como hay muchos tipos distintos, lo mejor es preguntar en la tienda cuál es cada uno (figura 4).

El Hardware, un mundo donde el temple profesional se une a la preparación teórica más firme.

La placa

Al principio habíamos dicho que era el sitio donde viven los componentes. Hace años no se utilizaban, sino que se conectaban mediante cables. Lo que pasa es que es más cómodo y seguro usar placas.

Suelen ser de baquelita o de fibra de vidrio, y llevan una cara recubierta de cobre. También las hay de doble cara y multicapa, y otras cosas raras que no nos interesan ahora. A la cara de cobre se le llama «cara de circuito» y es donde se hacen las conexiones. La otra es la de componentes, que es donde se ponen las resistencias, los condensadores, etc., y se ponen ahí para evitar que molesten al otro lado (ver figura 5).

Si nos queremos hacer una, el proceso más barato y sencillo es el siguiente:

— Limpiamos bien la cara de cobre.

— Una vez seca, pintamos con un rotulador resistente al agua aquellas partes que queremos que sigan cubiertas de cobre (es decir, las conexiones).

— Sumergimos la placa en una disolución de cloruro férrico (se encuentra en tiendas de electrónica) y, si puede ser, preferiblemente caliente y en un recipiente que no se utilice para otra cosa, porque los estropea bastante. Al cabo de un rato, más bien largo, se habrá desprendido el cobre que no nos interesa.

— Quitamos los restos de rotulador y taladramos la placa en los puntos donde queremos colocar los componentes. Ya está hecha la placa.

También se pueden utilizar otros productos en lugar de cloruro férrico, pero son más caros y peligrosos. Otra posibilidad es comprar placas ya hechas (tipo uniprint) que ya tienen hechos contactos, de forma que sólo hay que soldar los componentes y unirlos por los cables.

Soldar

Para soldar componentes se emplea un soldador tipo «lápiz» de unos 25 vatios de potencia, que no son caros. No es conveniente usar los soldadores de pistola, suelen ser demasiado potentes y pueden dañar los componentes.

Algo que no hay que olvidar es que el soldador se calienta lo suficiente para producir quemaduras serias, o provocar incendios, así que interesa tener un soporte adecuado para él, y no dejarlo nunca enchufado, ni cogerlo por la punta o apoyarse en él. En cuanto al estaño, es un compuesto de estaño, plomo y resinas para facilitar la fusión.

Lo primero que hay que hacer es, con el soldador caliente, fundir un poco de estaño sobre la punta del soldador, de forma que ésta quede estañada y protegida. Hay

que evitar rayar la punta, o cepillarla o limarla cuando esté caliente.

Para soldar componentes a la placa, calentamos la patilla del componente con el soldador y aplicamos el estaño al terminal, no al soldador. El estaño fundido debe quedar fluido, y solidificarse sin empañarse o de forma que quede con partes «distintas». Eso puede ser una mala soldadura.

Cuando se suelden transistores hay que tener cuidado de que no se calienten demasiado, pues podrían dañarse. Para evitarlo, además de hacer rápidamente la soldadura, es conveniente coger el terminal que estemos soldando con unos alicates de punta fina, de forma que sirvan para «radiar» el calor. En cuanto a los integrados, en ocasiones se usan «zócalos» para poder cambiar un integrado averiado sin necesidad de utilizar el soldador. La idea es soldar el zócalo y luego «enchufar» el integrado en el zócalo. Por supuesto, la sustitución de integrados o de cualquier otro componente debe hacerse *siempre* cuando el circuito esté desconectado, sin tensión ni corriente.

Ejemplo

Como ejemplo de todo lo visto, vamos a diseñar una fuente sencilla para alimentar un circuito con puertas lógicas TTL. El circuito aparece en la figura 6. Los elementos que aparecen en él (están numerados) son:

1. Transformador: su misión es que los 220 voltios de la red pasen a ser los 9 que necesitamos.

2. Puente de diodos: en lugar de usar 4 diodos separados, usamos 4 juntos. Las dos patillas con «~» se conectan a las salidas del transformador. El (+) se conecta al condensador C1, y el (–) será la masa de todo el conjunto.

3. Condensador: su misión es hacer que la señal sea algo más continua y dar corriente.

4. Regulador: éste es el que hace todo el trabajo sucio, y es un circuito integrado.

5. Condensador de salida: es para mejorar el funcionamiento.

Datos de la fuente: tensión de salida: 5 V; corriente máxima: 0.400 A.

Componentes: 1 transformador de 220 V a 9 V y 0,720 A; 1 puente rectificador; corriente: 0,5 A (para más seguridad), tensión: 20 voltios.

Condensadores: C1: 100 F, 16 voltios (ó 25); C2: 0,01 F, 9/16 voltios. Los dos son electrolíticos.

1 regulador integrado LM78M05 (en la figura se da el patillaje).

Francisco López Larrio

INFINITA PRECIS

Convocamos en este número a todos aquellos que amen las matemáticas, o quieran aprender algo sobre números gracias a su Spectrum. Vamos a analizar los sistemas artificiales de manejo de números enteros con artimética de infinita precisión.

La aritmética humana: contando con los dedos

Para poder enseñar a un ordenador a contar bien (qué horrendo es que 40000+40000 valga 8E4), primero hemos de comprender cómo lo hacemos NOSOTROS.

Cuando los humanos utilizamos números largos, ponemos una cifra detrás de otra hasta rellenar cadenas muy largas de dígitos que van del 0 al 9. Esto viene determinado naturalmente porque tenemos un total de diez dedos en las manos, y nos es muy fácil representar una cifra por medio de dichos diez dedos.

Un ordenador, sin embargo, cuenta con «dedos» que pueden ponerse en una o dos posiciones, los *bits*, usando la aritmética binaria. Si los hombres condideràsemos las dos posiciones que pueden tener los dedos, doblados o extendidos, podríamos contar desde 0 hasta 1023. Pero nuestro cerebro es demasiado complejo para ello, y por eso prefiere el sistema de contar el número de dedos extendidos.

Si queremos que un ordenador cuente en números muy grandes, nos bastaría con dividir estos en potencias de 256, con lo que podríamos asignar a cada cifra «hexaquincuagesimobicentesimal» un caracter del conjunto ASCII. Pero esto presenta un problema curioso: para tomar un número de teclado, o imprimir un número en la pantalla, podríamos perder muchas veces más tiempo que haciendo una división, si es que llegásemos a desarrollar el sistema adecuado para hacerlo.

Es por ello que nos resulta más cómodo utilizar tiras de caracteres en los que cada caracter represente exactamente la cifra que sea. Es decir, la tira "1234" representará el número 1234. De este modo es muy fácil intercambiar datos entre el ordenador y nosotros, y, como veremos, es igualmente fácil operar con estos números, ya que lo haremos exactamente igual que lo hacemos nosotros al usar lápiz y papel para sumar, restar, multiplicar o dividir.

Pasamos, pues, a explicar cada una de las operaciones elementales en nuestra aritmética para ordenador.

La suma y la resta: poner un número encima de otro

Como ya hemos dicho antes, el modo de operar con nuestros números largos es el mismo modo en que lo hacemos con un lápiz y papel. Por tanto, hagamos una suma.

Vamos a sumar 12345+567890. Primero escribimos un número encima del otro, haciendo coincidir las cifras de igual peso (figura 1).

A continuación sumamos las primeras cifras a estribor (por la derecha) (figura 2).

Luego la siguiente (figura 3).

El uno que nos ha sobrado lo

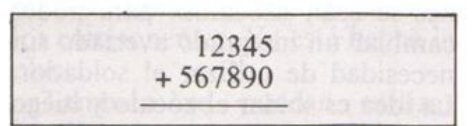


Figura 1

```
  12345
+ 567890
--------
       5
```
Figura 2

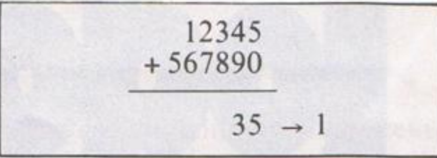


Figura 3

Listado 1

```
2000 IF LEN a$<LEN b$ THEN  LET
s$=a$: LET a$=b$: LET b$=s$
2010  LET l1=LEN a$: LET c$="":
LET l=LEN b$: LET a2=0: FOR i=0
TO l-1
2020 LET a1=VAL a$(l1-i)+VAL b$(
l-i)+a2: LET a2=INT (a1/10): LET
 c$=STR$ (a1-10*a2)+c$: NEXT i
2030  FOR i=l TO l1-1: LET a1=VA
L a$(l1-i)+a2: LET a2=INT (a1/10
): LET c$=STR$ (a1-a2*10)+c$: NE
XT i: IF a2 THEN  LET c$="1"+c$
2040  RETURN
```

Listado 2

```
3000 LET l1=LEN a$: LET l=LEN b$
: IF l1<l OR (l1=l AND a$<b$) TH
EN  LET c$="E": RETURN
3010 LET a2=0: LET c$="": FOR i=
0 TO l-1
3020 LET a1=VAL a$(l1-i)-VAL b$(
l-i)-a2: LET a2=a1<0: LET c$=STR
$ (a1+10*a2)+c$: NEXT i
3030 FOR i=l TO l1-1: LET a1=VAL
 a$(l1-i)-a2: LET a2=a1<0: LET c
$=STR$ (a1+10*a2)+c$: NEXT i
3040 FOR i=1 TO LEN c$-1: IF c$(
1)<>"0" THEN  RETURN
3050 LET c$=c$(2 TO ): NEXT i: R
ETURN
```

ION *o casi* EN BASIC

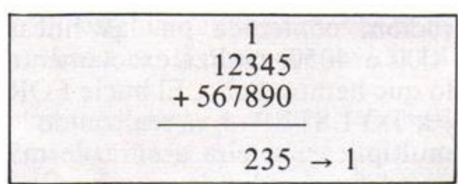


Figura 4

```
  12345
+567890
-------
 580235
```

Figura 5

metemos en la siguiente columna para sumarlo también (figura 4).

Y así sucesivamente, llegando al resultado (figura 5).

Evidentemente, si al sumar las últimas cifras nos hubiese sobrado otro uno, se lo habríamos añadido al resultado por la izquierda.

Cuando queremos que el ordenador calcule una suma nos encontramos con que hay tres partes distintas a tratar, según se ve en el listado 1:

a)La suma de las parejas de cifras coincidentes más lo que nos sobrase de la suma de la pareja anterior (líneas 2010 y 2020).

b) La suma de las cifras más altas del número más largo (las que no tienen pareja en el otro número) más lo que nos sobrase de la suma anterior (línea 2030).

c) Añadir lo que sobrase de las etapas anteriores como primera cifra del resultado (IF a2 THEN LET c$="1"+c$).

Todo ello lo podemos ver resuelto en el listado 1, que es una subrutina que suma el número contenido en a$ con el contenido en b$, devolviendo el resultado en c$.

En el listado 2 podemos ver cómo se resuelve la resta por un sistema similar. Sólo hemos de hacer dos observaciones al respecto. Como estamos trabajando con números enteros positivos, intentar restar un número de otro menor ha de considerarse un error, y por ello la subrutina devuelve como resultado la tira "E". Este resultado también se utiliza en la división. Otro punto interesante a resaltar es que, antes de abandonar la rutina de resta, se eliminan todos los ceros a la izquierda del resultado, cóo haría cualquier persona decente.

La multiplicación: ir paso a paso hacia la izquierda

Siguiendo con el sistema de obligar al ordenador a que trabaje

Listado 3

```
4000 LET k1=LEN r$: LET k=LEN m$
: IF k1<k THEN  LET s$=m$: LET m
$=r$: LET r$=s$: LET a2=k: LET k
=k1: LET k1=a2
4010 LET c$="0": FOR j=k TO 1 ST
EP -1: LET a3=VAL m$(j)
4020 GO SUB 4500:OYF b$="0" THEN
  GO TO 4050
4030 LET a$=c$: FOR i=1 TO k-j:
LET b$=b$+"0": NEXT i: GO SUB 20
00
4050 NEXT j: RETURN
4500 LET a2=0: LET b$="": IF a3=
0 THEN  LET b$="0": RETURN
4510 FOR i=k1 TO 1 STEP -1: LET
a1=a3*VAL r$(i)+a2: LET a2=INT (
a1/10): LET b$=STR$ (a1-10*a2)+b
$: NEXT i
4520 IF a2 THEN  LET b$=STR$ a2+
b$
4530 RETURN
```

Listado 4

```
4500>LET a2=0:LET b$="":IF a3=0T
HEN LET b$="0":RETURN
4510 FOR i=k1 TO 1 STEP -1: LET
a1=a3*VAL r$(i)+a2: LET a2=INT (
a1/10): LET b$=STR$ (a1-10*a2)+b
$: NEXT i
4520 IF a2 THEN  LET b$=STR$ a2+
b$
4530 RETURN
5000 LET k=LEN m$: LET k1=LEN r$
: LET n$="": IF k<k1 OR (k=k1 AN
D m$<r$) THEN  LET n$="0": RETUR
N
5010 LET a$=m$( TO k1): LET m$=m
$(k1+1 TO )
5020 GO SUB 5500: LET k=LEN m$:
IF k=0 THEN  GO TO 5040
5030 LET a$=a$+m$( TO 1): LET m$
=m$(2 TO ): GO TO 5020
5040 IF n$(1)="0" THEN  LET n$=n
$(2 TO ): GO TO 5040
5050 RETURN
5500 FOR n=9 TO 1 STEP -1: LET a
3=n: GO SUB 4500: GO SUB 3000: I
F c$="E" THEN  NEXT n: LET n=0:
LET c$=a$
5510 IF c$="0" THEN  LET c$=""
5520 LET a$=c$: LET n$=n$+STR$ n
5530 RETURN
```

como nosotros lo hacemos, vamos a descubrir cómo multiplicamos. Decimos descubrir porque la mayor parte de la gente no se da cuenta de cómo lo hace.

Cojamos dos números largos, por ejemplo el 3452 y el 5606721. El primer paso que realizamos es poner sobre el papel los dos núme-

```
  5606721
×    3452
---------
```

Figura 6

ros, uno debajo del otro con el más corto debajo (figura 6).

El sistema que utilizamos es ir multiplicando cada una de las cifras del número de abajo por el número de arriba. Cada vez que realizamos una de estas multiplicaciones, ponemos el número siguiente un paso más a la izquierda (figura 7).

```
     5606721
×       3452
------------
    11213442
   28033605
  22426884
 16820163
------------
```

Figura 7

Este avance hacia la izquierda es equivalente a multiplicar por 10 varias veces, como se ve en la figura 8.

Como colofón, sumamos dichos productos parciales, de lo que resulta la solución que buscábamos.

```
      5606721
×        3452
-------------
     11213442
    280336050
   2242688400
  16820163000
-------------
  19354400892
```

Figura 8

Antes hemos creado una rutina que sumaba dos números, por lo que modificaremos algo el proceso que realizamos sobre el papel. La rutina que escribamos ira sumando los productos parciales según los vaya generando, en vez de esperar al final para sumarlos todos juntos.

En cuanto a la multiplicación por un sólo dígito, podemos ver en el listado 3, en las líneas 4500 a 4530 la rutina que multiplica el número de una sola cifra contenido en la variable a3 por el número largo en r$, devolviendo el resultado en b$. Esta rutina es tan sencilla de entender que no daremos explicaciones más profundas.

La rutina principal de multiplicación, contenida en las líneas 4000 a 4050, realiza exactamente lo que hemos dicho. El bucle FOR j=k TO 1 STEP –1 va realizando la multiplicación cira a cifra de m$ por r$ por medio de sucesivas llamadas a la subrutina en 4500. El resultado, en b$, se suma al resultado anterior, en c$ por medio de la rutina de suma que creamos antes en lal la línea 2000. Así pues, la función que desempeña dicha rutina es devolver un valor c$=m$*n$.

La división por el sistema de prueba y sigue

El proceso de división que vamos a analizar es el de división entera, que es la que divide un número por otro devolviendo un cociente y un resto. Nosotros desarrollaremos una única rutina que devuelva ambos resultados, ya que es más cómodo.

```
4653253654 | 345
506          ----
```

Figura 9

LOS NUMEROS: EL ETERNO RETO MATEMATICO

Uno de los temas que más tinta ha hecho correr entre los teóricos matemáticos, es el de los números primos, números que presentan una serie de propiedades específicas que los hacen muy interesantes para aquellos que gustan de los problemas matemáticos incomprensibles.

Un número primo es un número natural (entero y positivo) que sólo puede ser dividido por sí mismo y por la unidad. Tal es el caso del número 11, y no lo es el 6, que es divisible por 2 y 3. Evidentemente, el primer asunto interesante relacionado con los primos que nos surge es el de identificarlos. Un primer modo de hacerlo es el de tomar el número que queremos saber si es primo o compuesto (no primo) y dividirlo entre todos los números menores que él para ver si entre dichas divisiones hay alguna que no deje resto. Este es un método burdo y basto, ya que sabemos que si un número es par no puede dividir a uno impar, por ejemplo. Investigando sobre esta idea, los matemáticos llegaron antiguamente a conclusiones interesantes para saber si un número es divisible por una serie concreta de ellos. Cabe contar entre dichas conclusiones que un número es divisible por dos si su última cifra es par, es divisible por tres si la suma de sus cifras también lo es, es divisible por cinco si su última cifra es 0 ó 5, etcétera.

Otro problema curioso fue el de elaborar una lista de números primos. El primero en dar una solución sencilla y elegante fue el griego Eratostenes, que inventó la «criba» de su mismo nombre. Esta consiste en escribir una lista de números del 1 al número más grande que queramos estudiar. Empezando por el dos, contaremos a partir del número siguiente de dos en dos, tachando cada vez el número en el que se acabe la cuenta. A continuación se toma el siguiente no tachado y se cuenta en la cantidad que él indique, tachando de nuevo los números en que paremos. De este modo, cuando no queden más números sin tachar a continuación del último utilizado, los que nos queden sin rayajo encima formarán la lista de primos buscada. Este método, realizado con lápiz y papel, deja de ser práctico

Para dividir un número por otro, ponemos el dividendo y, a continuación, el divisor metido en una caja (figura 9).

El sistema que seguimos es el de probar a ver si podemos dividir las

```
4653253654 | 345
-345       |----
-----          1
 120
```

Figura 10

primeras cifras del dividendo, en un número igual a las del divisor, por este último. Si es posible, se realiza la división y el cociente se pone bajo el divisor, como se muestra en la figura 10.

Esto se repite con el número formado por las cifras del resto anterior y las cifras no utilizadas hasta el momento (figura 11).

Para hallar estas divisiones intermedias nos basamos en la búsqueda de un número entre 0 y 9 tal que multiplicado por el divisor de un número menor que las cifras seleccionadas y tal que el divisor multiplicado por el siguiente número sea mayor que ellas. El ser humano, que es muy intuitivo, utiliza algoritmos aparentemente sencillos para hallar rápidamente el número de una cifra que precisamos, pero esto es difícil de imitar por un ordenador. Por ello, en él usaremos un sistema de búsqueda iterativo, en el que probaremos con los números del 9 al 1 en este orden hasta que encontremos el adecuado. Para ello, habremos de probar con cada uno a multiplicarlo por el divisor, por lo que haremos uso de la subrutina en la línea 4500 relacionada con la multiplicación.

```
4653253654 | 345
-345       |-----------
-----------  13487691 = cociente
1203253654
-1035
-----------
168253654
-1380
-----------
30253654
-2760
-----------
2653654
-2415
-----------
238654
-2070
-----------
31654
-3105
-----------
604
-345
-----------
RESTO = 259
```

Figura 11

A continuación habremos de ver el resultado de restar dicho producto de las cifras seleccionadas del dividendo, para lo que usaremos la rutina de resta en 3000. El valor devuelto "E" por ella nos servirá para probar si el número de una cifra elegido es válido, o hay que probar con otro.

En el listado 4 podemos ver la rutina completa de división, que realiza la de m$ entre r$, dejando el cociente en n$ y el resto en c$. Se ha dividido en dos bloques, para su mejor comprensión. El primero, líneas 5000 a 5040, se encarga de la repetitiva labor de ir elaborando el resultado a partir de lo que le devuelve la rutina en las líneas 5500 a 5530, que es la encargada de realizar cada una de las pequeñas divisiones parciales.

a partir del número 100, y no resulta interesante para saber si un número concreto es primo.

La siguiente cuestión que se planteó es una consecuencia lógica del uso de la criba de Eratostenes. Si hubiese un número finito de primos, resultaría interesante hacer una criba todo lo grande que se precisase y elaborar una lista de primos que se publicaría con tapas de oro (es un decir). Fue el famoso Euclides el que demostró de un modo definitivo que el conjunto de primos es infinito, lo que echó al traste las esperanzas de muchos matemáticos (y editores). La prueba es muy sencilla, y está basada en que, si sumamos 1 a un número A que es múltiplo dé otro número B, el número resultante, C, no es múltiplo de B (salvo si B es la unidad). Así pues, supongamos que disponemos de una lista de primos. Multiplicando todos entre sí (salvo el 1), nos da un número que es múltiplo de todos ellos. Sumando la unidad a este resultado, tendremos otro número que no es múltiplo de ninguno de ellos. Este número puede ser primo, en cuyo caso lo unimos a la lista. Si no es primo, entonces existen al menos dos números primos menores que él, que le dividen, en cuyo caso los uniremos a la lista. Con esta ampliada de este modo, repetimos la operación, volviéndose a ampliar de nuevo. Como esto es cierto siempre, entonces la lista nunca estaría cerrada, y por tanto es infinita.

Ante tal prueba de fuerza, se decidió olvidar por completo la cuestión de conocer todos los primos que existen, y se plantearon dos nuevos problemas: descubrir algoritmos útiles y rápidos que permitan descubrir si un número es primo, problema que trasvasamos a los lectores por medio de este concurso; y descubrir números primos muy gordos, lo que en Estados Unidos hace furor (como todo lo que tiene que ver con «lo más grande», «lo más largo», etc.) entre los programadores de ordenadores. El récord actual, que sepamos en esta redacción, está en un número primo de 39.751 cifras de la forma 2 n–1, descubierto en la conocida Universidad de Harvard.



SHERLOCK

Melbourne House
Spectrum 48 K

Para todos aquellos que tienen vocación de detective y les apasionen los relatos de Agatha Christie y Sherlock Holmes, Philip Mitchell, ya conocido por ser el creador del famoso juego del «Hobbit» nos deleita ahora con una nueva aventura conversacional basada en las increíbles hazañas del inefable Sherlock.

El juego, al igual que el Hobbit, está desarrollado en inglés y permite al jugador conversar, hacer preguntas e incluso discutir las conclusiones que va presentando el programa, aunque esto, en principio, puede presentar algunos inconvenientes a los jugadores que no dominen bien el inglés.

El objetivo del juego como podéis suponer es descubrir al asesino de unos crímenes, para ello tendrás que tener las ideas muy claras, interrogar a sospechosos, hacer deducciones y estudiar las pistas que vayas encontrando.

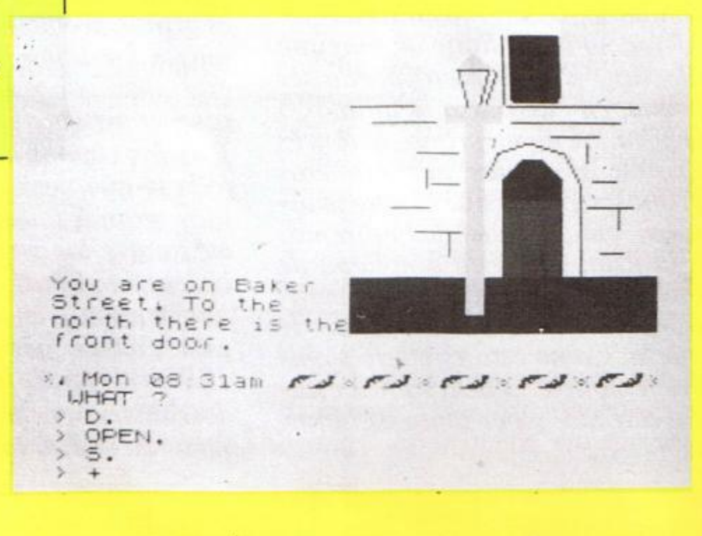
La pantalla aparece dividida en dos zonas, en la parte superior va apareciendo el texto que explica lo que ocurre en ese momento y los gráficos, y la parte inferior (separadas por un reloj que indica el tiempo relativo en la aventura) es donde introduces tus preguntas y conclusiones.

Junto al programa viene también una hoja que puede resultar muy útil a la hora de deducir pistas, ya que se refiere a itinerarios del metro con horas de llegada y de salida a los lugares en los que se centra la aventura.

Las pantallas gráficas animan la acción del juego y nos sitúan en el ambiente en el que se desarrolla la acción, algunas veces incluso nos dan pistas o despiertan inquietudes sobre determinado hechos.

Cuando no tengas pistas o quieras consultar algo puedes recurrir a tu viejo amigo Watson que te dará información y que siempre te acompañará donde quiera que vayas.

El recibidor de la casa de Sherlock Holmes es el primer paso para empezar nuestra aventura.



Baker Street: la calle donde vive nuestro héroe y uno de los primeros sitios donde ponemos a prueba el ingenio.

Control: Teclado.

Jugadores: Uno.

Gráficos: Resultan atractivos al aparecer al lado del texto y situarnos en el lugar de los hechos.

Sonido: No tiene.

Nivel de dificultad: En general es bastante elevado.

Originalidad: Aunque ya existen precedentes similares en el mercado, la idea en la que se basa es original.

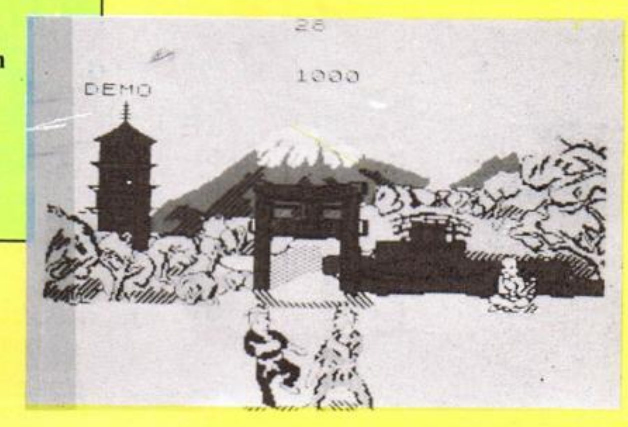
Conclusión: El juego es interesante pero puede resultar complicado sobre todos a aquellos que no dominen bien el inglés.

THE WAY OF THE EXPLODIGN FIST

ERBE
Spectrum 48 K
2.900 ptas.

El primer combate se realiza en el campo y constituye el desafío inicial de todo aspirante a karateka.

En los últimos tiempos los juegos de lucha están adquiriendo cada día un mayor auge en videojuegos y, dentro de éstos podemos encuadrar a FIST, un programa de kárate con un objetivo a seguir: El de derribar contrarios, sólo así podrás pasar a una categoría mayor y enfrentarte a otros contrincantes más hábiles y rápidos. Para conseguirlo, deberás tener un perfecto dominio tanto en los golpes como en la defensa si no quieres verte sorprendido por uno de los ataques del contrario que pueden ser a veces espectaculares y si alguno de los dos contendientes logra llegar antes de este tiempo a la puntuación de dos ipones (cada ipón se representa como una luna) vence el combate; el sistema de puntuación se realiza en función de la dificultad del golpe y varía de medio a un ipón.



Una vez dentro de la mansión del Buda siguen los combates, pero esta vez contra un contrincante más temible.

En el caso que quieran participar dos jugadores el programa ofrece la oportunidad de enfrentarlos, esto crea una gran rivalidad entre ellos y hace más interesante la competición. El juego puede ser controlado tanto por el teclado (que puede ser redefinido) como por joystick.

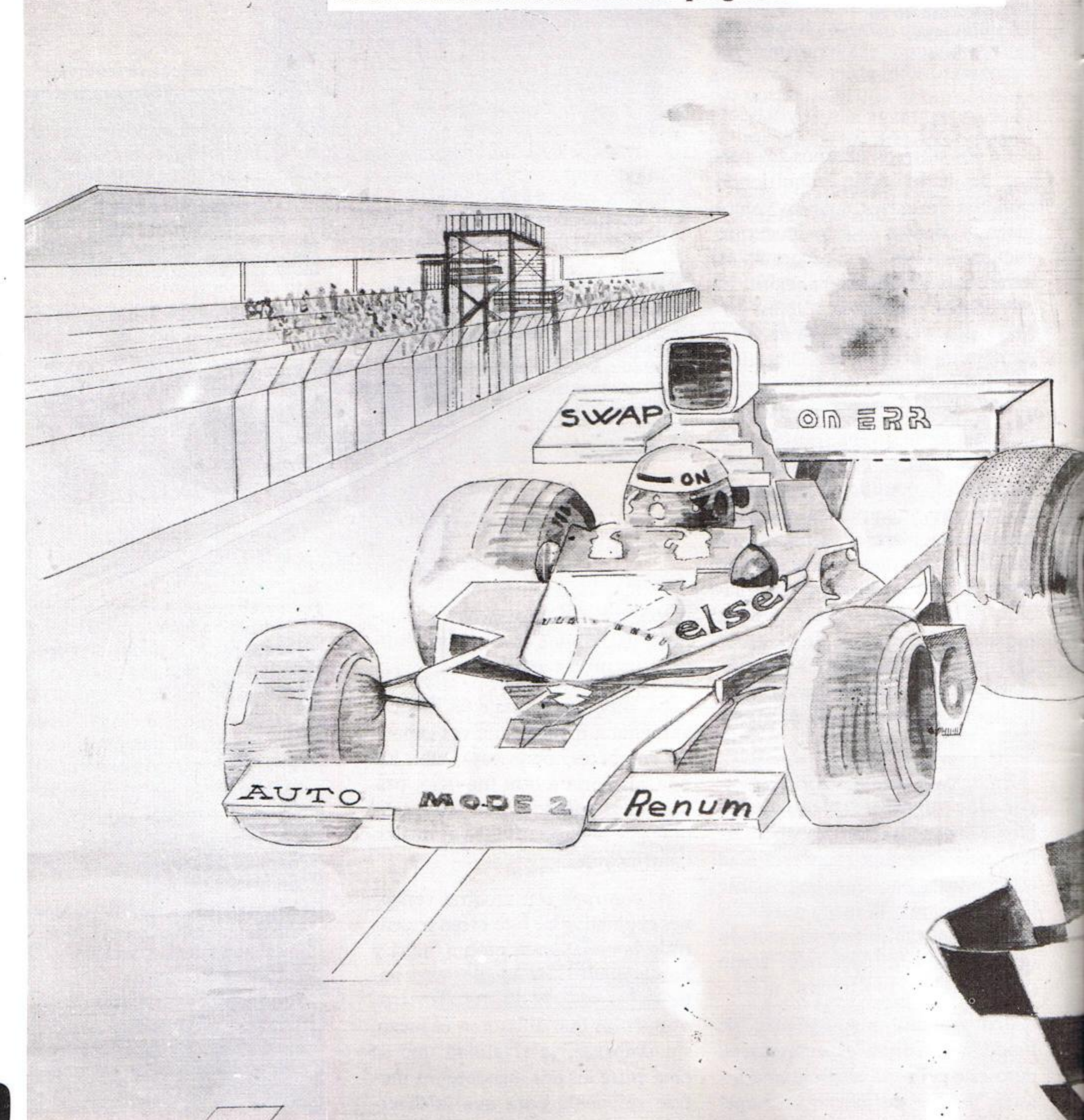
Los gráficos que aparecen de fondo no resultan espectaculares, pero esto no resta al programa interés, ya que no juegan un papel tan importante como la definición de los movimientos de los karatekas y la rapidez con que se realizan.

La dificultad, como es lógico, aumenta a medida que vas ganando combates, pero para subir de una categoría a otra (novicio, primer dan, segundo dan, etc.), tendrás que ganar dos veces al mismo contrincante.

Al contrario que en otras versiones el objetivo de este juego se centra solamente en la propia lucha y habilidad de cada jugador para batir al contrario y no incluye otros obstáculos que dificulten el juego, sin embargo, la rivalidad que se crea entre los dos jugadores es motivo suficiente para que la diversión esté asegurada.

Control: Teclado y Joystick.
Jugadores: Uno (contra el ordenador) o dos (contra ellos mismos).
Gráficos: No son muy brillantes, los dibujos de fondo varían con relación al nivel.
Sonido: Sólo se aprecia al pasar de pantalla o al efectuar un golpe.
Nivel de dificultad: Aumenta con la categoría.
Originalidad: Es bastante original aunque parece que en el mercado hay ya bastantes juegos de kárate.
Conclusión: Este es un buen juego de lucha que como rasgo más interesante es la posibilidad de enfrentar a dos jugadores.

En este mes se describe el último conjunto de rutinas internas que usa el nuevo sistema operativo, con lo que sólo nos queda definir el funcionamiento y uso de cada uno de los comandos, cosa que dejamos para el próximo mes, en el que acaba esta serie de artículos destinada a explicar el funcionamiento interno de este programa.



Un nuevo operativo para el Spectrum

④

El interprete BASIC es utilizado de dos formas, entrando en LISCAN realiza el análisis sintáctico de la línea, y entrando en LIRUN interpreta la línea o el programa. La forma de distinguir las dos funciones es mediante el bit 7 de la variable FLAGS, que está a uno durante la ejecución y a cero en el análisis sintáctico. El test de este bit se hace siempre llamando a SINTAX-Z en la dirección 2530. De esta forma se ahorra un byte en cada llamada, aunque es más lento.

Para interpretar una línea opera de la siguiente manera. En primer lugar coge el código del comando. Si éste corresponde a una función del BASIC estándar busca en la tabla TABCRD para ver si se trata de uno de los comandos que han sido rescritos. Si es así, se les asigna un código para entrar en la nueva tabla de sintaxis. Si es un comando nuevo o uno antiguo no re-

```
LISCAN RES   7,(IY+1)
       CALL  ELINO
       XOR   A
       LD    (^5C47),A
       DEC   A
       LD    (^5C3A),A
       JR    STMTL1
STLOOP CALL  NEXTCH
STMTL1 CALL  ^16BF
       INC   (IY+13)
       JP    M,^1C8A
       CALL  GETCH
       LD    B,^00
       CP    ^0D
       JP    Z,LINEND
       CP    ^3A
       JR    Z,STLOOP
       LD    HL,STMRET
       PUSH  HL
       LD    C,A
       CALL  NEXTCH
       LD    A,C
       CP    ^CE
       JR    NC,NNCOM
       SUB   ^18
       JR    C,REPC
       CP    ^08
       JR    C,NUCOM
REPC   JP    ^1C8A
COMRD  LD    A,(HL)
NUCOM  LD    HL,TABSIN
       JR    GNPAR
NNCOM  LD    HL,TABCRD
       LD    C,A
       CALL  ^16DC
       JR    C,COMRD
       LD    A,C
       LD    HL,^1A48
       SUB   ^CE
GNPAR  LD    C,A
       ADD   HL,BC
       LD    C,(HL)
       ADD   HL,BC
       JR    GETPAR
SCLOOP LD    HL,(^5C74)
GETPAR LD    A,(HL)
       INC   HL
       LD    (^5C74),HL
       LD    BC,SCLOOP
       PUSH  BC
       LD    C,A
       CP    ^20
       JP    NC,SEPARA
       CP    ^03
       JR    NZ,NOCL3
       RST   ^18
       CALL  ^1CDE
CLASSO CP    A
CLASS5 POP   BC
       CALL  Z,SHEEND
       EX    DE,HL
       JP    ^1C16
NOCL3  AND   A
       JR    Z,CLASSO
       CP    ^05
       JR    NZ,NOCL5
       DEC   B
       JR    CLASS5
NOCL5  CP    ^02
```

definido se reduce su rango para poder entrar en las tablas y se inicializa HL al comienzo de la tabla correspondiente. A partir de este momento todos los comandos se tratan de la misma forma, aunque utilizando la tabla de la ROM o la nueva según corresponda. De esta tabla se extrae un dato que una vez sumado a la dirección en que se encuentra apunta al comienzo de los parámetros correspondientes a ese comando. Estos parámetros incluyen la clase o clases a que pertenece el comando, los separadores, si es que los necesita, y por último la dirección en que se encuentra la rutina de ese comando.

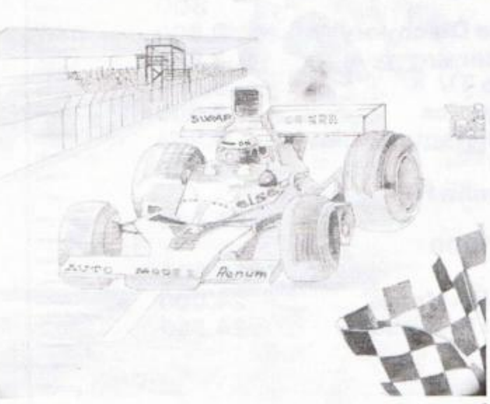
Al interpretar una línea coge un código de comando, si este es uno de los del Spectrum, mira a ver si ha sido redefinido.

La clase de un comando indica qué es lo que debe haber detrás del comando. Por ejemplo, CLS es un comando de clase 0, lo que quiere decir que no debe haber nada más detrás de él. Si hubiera algo más se produciría un error de sintaxis (Nonsense in BASIC). Un comando puede utilizar varias clases. Por ejemplo, los parámetros correspondientes a FOR son: 04 3D 06 CC 06 05 03 1D. 04 indica que es de clase 4, debe encontrar el nombre de una variable de una sola letra. 3D es el código de '=', caracter que debe estar detrás de la varia-

```
        JR    NZ,NOCL2
        POP   BC
        CALL  ^1C56
        CALL  CHEEND
        RET
NOCL2   CP    ^07
        JP    NZ,^1B63
        BIT   7,(IY+1)
        RES   0,(IY+2)
        CALL  NZ,^0D4D
        POP   AF
        LD    A,(^5C74)
        SUB   ^13
        CALL  ^21FC
        CALL  CHEEND
        JP    ^1CAD
STMRET  CALL  ^1F54
        JP    NC,^1B7B
        BIT   7,(IY+10)
        JR    NZ,STNEXT
        LD    HL,(^5C42)
        BIT   7,H
        JR    Z,LINEW
LIRUN   LD    HL,^FFFE
        LD    (^5C45),HL
        LD    HL,(^5C61)
        DEC   HL
        LD    DE,(^5C59)
        DEC   DE
        LD    A,(^5C44)
        JR    NEXTLI
LINEW   CALL  ^196E
        LD    A,(^5C44)
        JR    Z,LIUSE
        AND   A
        JR    NZ,REPN
        LD    B,A
        LD    A,(HL)
        AND   ^C0
        LD    A,B
        JR    Z,LIUSE
        RST   8
        DEFB  ^FF
REM     POP   BC
LINEND  CALL  ^2530
        RET   Z
        LD    HL,(^5C55)
        LD    A,^C0
        AND   (HL)
        RET   NZ
        XOR   A
LIUSE   CP    ^01
        ADC   A,^00
        LD    D,(HL)
        INC   HL
        LD    E,(HL)
        LD    (^5C45),DE
        INC   HL
        LD    E,(HL)
        INC   HL
        LD    D,(HL)
        EX    DE,HL
        ADD   HL,DE
        INC   HL
NEXTLI  LD    (^5C55),HL
        EX    DE,HL
        LD    (^5C5D),HL
        LD    D,A
        LD    E,^00
        LD    (IY+10),^FF
        DEC   D
        LD    (IY+13),D
        JP    Z,STLOOP
        INC   D
        CALL  EASTMT
        JR    Z,STNEXT
REPN    RST   8
        DEFB  ^16
CHEEND  CALL  ^2530
        RET   NZ
        POP   BC
        POP   BC
STNEXT  RST   ^18
        CP    ^0D
        JR    Z,LINEND
        CP    ^3A
GSTLO   JP    Z,STLOOP
        LD    A,C
        AND   A
        JR    Z,GSTLO
        RST   8
        DEFB  ^0B
TABCRD  DEFB  ^F7,^08;RUN
        DEFB  ^FB,^09;CLS
        DEFB  ^FD,^0A;CLEAR
        DEFB  ^EA,^0B;REM
        DEFB  ^FA,^0C;IF
        DEFB  ^E6,^0D;NEW
        DEFB  ^EB,^0E;FOR
        DEFB  ^F5,^0F;PRINT
        DEFB  ^EE,^10;INPUT
        DEFB  ^E9,^11;DIM
        DEFB  ^F0,^12;LIST
        DEFB  ^E3,^13;READ
        DEFB  ^E4,^14;DATA
        DEFB  ^FC,^15;DRAW
        DEFB  ^E0,^16;LPRINT
        DEFB  ^E1,^17;LLIST
        DEFB  ^D8,^18;CIRCLE
        DEFB  ^CE,^19;DEFFN
        DEFB  ^00
TABSIN  DEFB  ^61,^6D,^57
        DEFB  ^5A,^60,^62
        DEFB  ^64,^6A,^12
        DEFB  ^14,^16,^18
        DEFB  ^1A,^1E,^20
        DEFB  ^27,^29,^2B
        DEFB  ^2D,^2F,^31
        DEFB  ^33,^36,^38
        DEFB  ^3A,^3D
        DEFB  ^03
        DEFW  RUN
        DEFB  ^00
        DEFW  CLS
        DEFB  ^03
        DEFW  CLEAR
        DEFB  ^05
        DEFW  REM
        DEFB  ^06,^CB,^05
        DEFW  IF
        DEFB  ^00
        DEFW  NEW
```

En la tabla de comandos se indican los parámetros que deben ir detrás de cada instrucción, de este modo se puede comprobar si la sintaxis es correcta.

```
        DEFB ^04,"=",^06
        DEFB ^CC,^06,^05
        DEFW FOR
        DEFB ^05
        DEFW PRINT
        DEFB ^05
        DEFW INPUT
        DEFB ^05
        DEFW DIM
        DEFB ^05
        DEFW LIST
        DEFB ^05
        DEFW READ
        DEFB ^05
        DEFW DATA
        DEFB ^09,^05
        DEFW DRAW
        DEFB ^05
        DEFW LPRINT
        DEFB ^05
        DEFW LLIST
        DEFB ^09,^05
        DEFW CIRCLE
        DEFB ^05
        DEFW DEFFN
        DEFB ^08,^00
```




```
        DEFW AUTO
        DEFB ^08,^05
        DEFW RENUM
        DEFB ^05
        DEFW ELSE
        DEFB ^05
        DEFW ONAGO
        DEFB ^05
        DEFW SWAP
        DEFB ^06,^00
        DEFW MODE
        DEFB ^05
        DEFW ONERR
        DEFB ^00
        DEFW COMND
RUN     CALL ^1E67
        LD   BC,^0000
        CALL ^1E45
        JR   CLERUN
CLEAR   CALL ^1E99
CLERUN  LD   A,B
        OR   C
        JR   NZ,CLEAR1
        LD   BC,(^5CB2)
CLEAR1  PUSH BC
        LD   DE,(^5C4B)
        LD   HL,(^5C59)
        DEC  HL
        CALL ^19E5
        CALL CLS
        JP   ^1EC6
IF      POP  BC
        CALL ^2530
        JR   Z,IF1
        RST  ^28
        DEFB ^02,^38
        EX   DE,HL
        CALL ^34E9
        JP   C,LOELSE
IF1     CALL GETCH
        JP   STMTL1
LOELSE  LD   B,(IY+13)
        LD   C,^01
CBELSE  CALL GETCH
        CP   ^18
        JR   NZ,NELSE
        DEC  C
        JR   Z,CELSE
NELSE   CP   ^FA
        JR   NZ,NOIF
        INC  C
NOIF    INC  B
        CALL LONST
        CALL ^0074
        JR   NC,CBELSE
        JP   LINEND
CELSE   LD   (IY+13),B
        CALL NEXTCH
        JP   STMTL1
LONST   LD   A,(HL)
        CALL TESTC
        CALL ^18B6
        LD   (^5C5D),HL
        CP   ^3A
        RET  Z
        CP   ^CB
        RET  Z
        CP   ^0D
        SCF
        RET  Z
        INC  HL
        JR   LONST
TESTC   CP   ^22
        RET  NZ
OTRC    INC  HL
        LD   A,(HL)
        CP   ^22
        RET  Z
        CP   ^0D
        JR   NZ,OTRC
        POP  AF
        SCF
        RET
FOR     RST  ^18
        CP   ^CD
        JR   NZ,FUSE1
        RST  ^20
        CALL ^1C82
        CALL CHEEND
        JP   ^1D16
FUSE1   CALL CHEEND
        JP   ^1D13
LPRINT  LD   A,^03
        JR   PRINT1
PRINT   LD   A,^02
PRINT1  CALL ^2530
        CALL NZ,^1601
        CALL ^0D4D
        CALL ^1FDF
        CALL CHEEND
        RET
INPUT   CALL ^2530
        JR   Z,INP1
        LD   A,^01
        CALL ^1601
        CALL CLSLO
INP1    LD   (IY+2),^01
        CALL INIT1
        CALL CHEEND
        LD   BC,(^5C88)
        LD   A,(^5C6B)
        CP   B
        JR   C,INP2
        LD   C,^21
        LD   B,A
INP2    LD   (^5C88),BC
        LD   A,^19
        SUB  B
        LD   (^5C8C),A
        RES  0,(IY+2)
        CALL ^0DD9
        JP   CLSLO
INIT1   CALL ^204E
        JR   Z,INIT1
        CP   ^28
        JR   NZ,INIT2
        RST  ^20
        CALL ^1FDF
        RST  ^18
        CP   ^29
        JP   NZ,^1C8A
        RST  ^20
        JP   INNE2
INIT2   CP   ^CA
        JR   NZ,INIT3
        RST  ^20
        CALL ^1C1F
        SET  7,(IY+55)
        BIT  6,(IY+1)
        JP   NZ,^1C8A
        JR   INPROM
INIT3   CALL ^2C8D
        JP   NC,INNE1
        CALL ^1C1F
```

```
                RES   7,(IY+55)
        INPROM  CALL  ^2530
                JP    Z,INNE2
                CALL  ^16BF
                LD    HL,^5C71
                RES   6,(HL)
                SET   5,(HL)
                LD    BC,^0001
                BIT   7,(HL)
                JR    NZ,INPR2
                LD    A,(IY+1)
                AND   ^40
                JR    NZ,INPR1
                LD    C,^03
        INPR1   OR    (HL)
                LD    (HL),A
        INPR2   RST   ^30
                LD    (HL),^0D
                LD    A,C
                RRCA
                RRCA
                JR    NC,INPR3
                LD    A,^22
                LD    (DE),A
                DEC   HL
                LD    (HL),A
        INPR3   LD    (^5C5B),HL
                BIT   7,(IY+55)
                JR    NZ,INVAR3
                LD    HL,(^5C5D)
                PUSH  HL
                LD    HL,(^5C3D)
                PUSH  HL
        INVAR1  LD    HL,INVAR1
                PUSH  HL
                BIT   4,(IY+48)
                JR    Z,INVAR2
                LD    (^5C3D),SP
        INVAR2  LD    HL,(^5C61)
                CALL  ^11A7
                LD    (IY+0),^FF
                CALL  EDITOR
                RES   7,(IY+1)
                CALL  ^21B9
                JR    INVAR4
        INVAR3  CALL  EDITOR
```

Una vez acabada de analizar una sentencia se comprueba la tecla BREAK y pasa a STÑEXT para analizar la siguiente.

```
        INVAR4  LD    (IY+34),^00
                CALL  ^21D6
                JR    NZ,INVAR5
                CALL  EDCOPY
                LD    BC,(^5C82)
                CALL  ^0DD9
        INVAR5  LD    HL,^5C71
                RES   5,(HL)
                BIT   7,(HL)
                RES   7,(HL)
                JR    NZ,INVAR6
                POP   HL
                POP   HL
                LD    (^5C3D),HL
                POP   HL
                LD    (^5C5F),HL
                SET   7,(IY+1)
                CALL  ^21B9
                LD    HL,(^5C5F)
                LD    (IY+38),^00
                LD    (^5C5D),HL
                JR    INNE2
        INVAR6  LD    HL,(^5C63)
                LD    DE,(^5C61)
                SCF
                SBC   HL,DE
                LD    B,H
                LD    C,L
                CALL  ^2AB2
                CALL  ^2AFF
                JR    INNE2
        INNE1   CALL  ^1FFC
        INNE2   CALL  ^204E
                JP    Z,INIT1
                RET
        DIM     CALL  ^28B2
                JP    NZ,^1C8A
                CALL  ^2530
                JR    NZ,DRUN
                RES   6,C
                CALL  ^2996
                CALL  CHEEND
        DRUN    JP    ^2C15
        LLIST   LD    A,^03
                JR    LIST1
        LIST    LD    A,^02
        LIST1   LD    (IY+2),^00
                CALL  ^2530
                CALL  NZ,^1601
                RST   ^18
                CALL  ^2070
                JR    C,LIST4
                RST   ^18
                CP    ^3B
                JR    Z,LIST2
                CP    ^2C
                JR    NZ,LIST3
        LIST2   RST   ^20
                CALL  ^1C82
                JR    LIST5
        LIST3   CALL  ^1CE6
                JR    LIST5
        LIST4   CALL  ^1CDE
        LIST5   CALL  CHEEND
                JP    ^1825
        DATA    RST   ^18
                CALL  ^2530
                JP    NZ,^1E37
        DATA1   CALL  ^24FB
                CP    ^2C
                CALL  NZ,CHEEND
                RST   ^20
                JR    DATA1
        DRAW    RST   ^18
                CP    ^2C
                JR    Z,DR3P
                CALL  CHEEND
```

GUSANEZ

por José C. Tomás

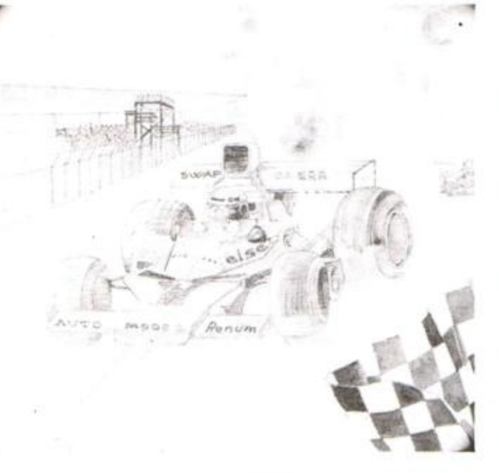






ble. Clase 06 indica una expresión numérica, CC es el código de 'TO', otra vez clase 6, otra expresión. Por último clase 5, indica que el resto del análisis lo hace la rutina del comando, cuya dirección indican los dos últimos bytes.

Los comandos que acaban en clase 5 son llamados tanto si está analizando como si está ejecutando una línea. Todos ellos en un momento dado llaman a CHECK-END, que sólo devuelve el control si está en ejecución. Si no continúa



directamente en STMT-NEXT en la ROM. Por esta razón se ha reproducido estos comandos hasta el punto en que llaman a CHECK-END.

Los comandos de clase 0B tienen el mismo problema, sin embargo no han sido reproducidos

El juego de caracteres es el mismo que usa el Tasword y se puede sacar de este programa con un sencillo conjunto de instrucciones, lo que ahorra bastante trabajo.

por que alargarían demasiado el programa. Estos comandos son los correspondientes a las rutinas del cassette y el problema que presenta el no introducirlos no es muy grave, lo único que ocurre es que no se puede poner un comando nuevo detrás de una de estas instrucciones en la misma línea. El programa sería capaz de ejecutarlo, pero no puede superar el análisis sintáctico, ya que intenta analizar un comando nuevo con las rutinas de la ROM.

Una vez que ha acabado de analizar/ejecutar la sentencia comprueba la tecla BREAK en STMRET y pasa a STNEXT para analizar la siguiente sentencia o la siguiente línea.

Los caracteres

Para acabar resolvemos el problema de introducir el juego de caracteres. Vamos a dar dos métodos, uno a partir del programa Tasword Two y otro a partir de los listados del mes de septiembre.

Si lo queréis sacar del Tasword debéis cargarlo y volver al BASIC. Hacer NEW, con lo que únicamente se borrará el programa BASIC. A continuación teclear el programa siguiente:

```
10 FOR i=32000 TO 32015:READ
   a:POKE i,a:NEXT i
20 DATA 33,0,239,1,128,3,175,
   237,111,35,11,120,177,32,247,
   201.
30 RANDOMIZE USR 32000:
   SAVE "chr"CODE 61184,896
```

y grabar en una cinta el juego de caracteres. Por último carga el código producido por el ensamblador y encima el juego de caracteres utilizando: LOAD "chr"CODE 60358, y vuelve a grabar el programa.

Si lo quieres sacar del listado de DATAS del mes de septiembre debes teclear las líneas comprendidas entre la 1060 y la 1340. Teclea también las líneas 10 a 110 cambiando la 10 y la 30 por:

```
10 CLEAR 60168:LET n=60351:
   RESTORE
30 FOR i=1060 TO 1340 STEP 10
```

Ejecuta el programa y cuando acabe haz SAVE "chr"CODE 60358,896. Carga el programa, encima el juego de caracteres y vuelve a grabarlo.

```
                JP    ^2477
        DR3P    RST   ^20
                CALL  ^1C82
                CALL  CHEEND
                JP    ^2394
        CIRCLE  RST   ^18
                CP    ^2C
                JP    NZ,^1C8A
                RST   ^20
                CALL  ^1C82
                CALL  CHEEND
                JP    ^232D
        READ3   RST   ^20
        READ    CALL  ^1C1F
                CALL  ^2530
                JP    NZ,^1DF5
                RST   ^18
                CP    ^2C
                JR    Z,READ3
                CALL  CHEEND
                RET
        DEFFN   RST   ^18
                CALL  ^2530
                JP    NZ,^1F65
                SET   6,(IY+1)
                CALL  ^2C8D
                JR    NC,DFN4
                RST   ^20
                CP    ^24
                JR    NZ,DFN2
                RES   6,(IY+1)
                RST   ^20
        DFN2    CP    ^28
                JR    NZ,DFN7
                RST   ^20
                CP    ^29
                JR    Z,DFN6
        DFN3    CALL  ^2C8D
        DFN4    JP    NC,^1C8A
                EX    DE,HL
                RST   ^20
                CP    ^24
                JR    NZ,DFN5
                EX    DE,HL
                RST   ^20
        DFN5    EX    DE,HL
                LD    BC,^0006
                CALL  ^1655
                INC   HL
                INC   HL
                LD    (HL),^0E
                CP    ^2C
                JR    NZ,DFN6
                RST   ^20
                JR    DFN3
        DFN6    CP    ^29
                JR    NZ,DFN7
                RST   ^20
                CP    ^3D
                JR    NZ,DFN7
                RST   ^20
                LD    A,(^5C3B)
                PUSH  AF
                CALL  ^24FB
                POP   AF
                XOR   (IY+1)
                AND   ^40
        DFN7    JP    NZ,^1C8A
                CALL  CHEEND
```



Novedad: Dibujando con ratón

Archivos con Archive

Gáficos en Superbasic

NOTICIAS

DIBUJANDO CON RATON

En el SIMO ha hecho su presentación un nuevo programa para el QL que por sus características va a hacer feliz a más de uno, ya que se trata de un evolucionado paquete para la realización de dibujos. Se llama Star Mouse y ha sido desarrollado por la casa PURICORP de Guadalajara, hecho bastante infrecuente ya que la mayoría de los programas que se venden en nuestro país (y sobre todo si están en lenguaje máquina como éste) provienen de Inglaterra.

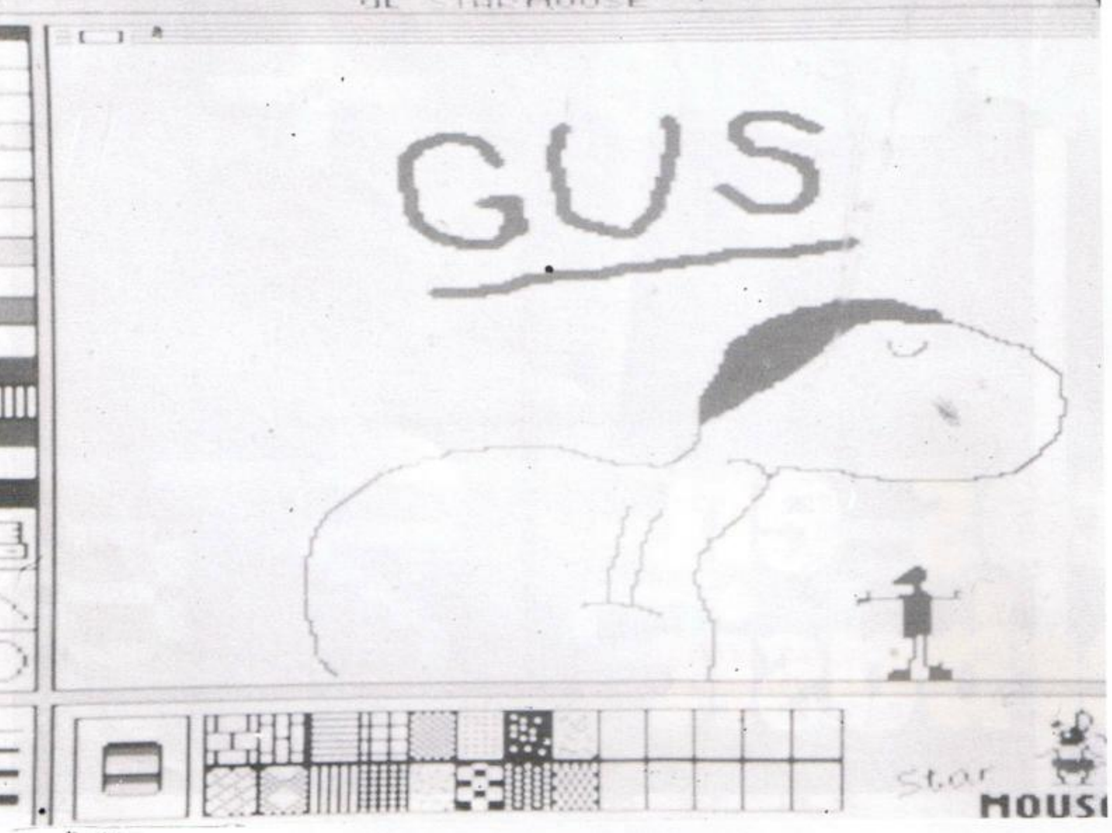
El paquete parte de una concepción distinta a la emplada hasta ahora, ya que emplea un dispositivo llamado «ratón» y es conocido desde hace mucho (fue inventado a principios de los 70 en los laboratorios de la Xerox en Palo Alto), no ha empezado a ser conocido masivamente hasta hace poco que algunos ordenadores personales lo han comenzado a usar.

Con este dispositivo se abandona el habitual sistema de movimiento a base de flechas de cursor y se sustituye por este dispositivo, que es mucho más intuitivo y más fácil de manejar, de modo que el tiempo empleado en la realización de los dibujos se acorta sensiblemente.

Al cargar el programa la mayor parte de la pantalla queda ocupada por el área de dibujo, blanca en un principio, a la izquierda de la cual aparece una serie de dibujos que representan los útiles que tenemos disponibles. Uno de ellos es una brocha, otro un cubo de pintura para rellenar áreas, otro realiza cuadrados, etcétera. También en este menú hay una serie de opciones adicionales, como un dibujo de una impresora que hace que lo que se muestra en pantalla salga impreso.

Debajo de ese menú hay otro de menor tamaño en el que se elige el tamaño de la línea con que se dibuja y por último, ocupando toda la parte inferior de la pantalla, se encuentra otro menú con las tramas disponibles y que usará la brocha, ya que ésta no sólo dibuja en negro, sino que puede hacerlo en diversos colores y tonos de gris.

Todas estas elecciones que hemos mencionado se hacen moviendo el ratón por encima de la mesa, lo que produce un movimiento equivalente a una pequeña flecha por la pantalla. Para elegir algo basta colocar la flecha encima y pulsar el botón, ejecutándose la acción. El proceso de dibujado se realiza con la misma sencillez y los efectos que se consiguen son realmente buenos, incluso para personas que no lo han manejado con anterioridad. Las perspectivas que tiene son bastante buenas, y una prueba de ello es que la casa Investrónica lo apoya y hemos podido verlo en su stand en el SIMO.



Nuestro simpático amigo Gusánez aparece en todas partes, incluso en el pantalla de dibujo del Star Mouse.

Recibos de Navidad

Ahora que vienen las Navidades, todos queremos repartir entre los familiares y amigos participaciones de la lotería de Navidad. Con un QL, una impresora y este programa, tendrás la posibilidad de personalizar estas participaciones y demostrar a más de un escéptico la utilidad de estas máquinas.

Generalidades

El programa está hecho para ser usado con una impresora CPA 80. Para otras deberán ponerse los códigos correspondientes en las siguientes líneas:

390: IMPRESION EXPANDIDA & LOT$ & IMPRESION NORMAL

1350: ESPACIADO ENTRE LINEAS A 1/8"

Si no tienes la versión española del QL, debes eliminar la línea 110, y respecto a las letras acentuadas, deberás elegir entre:

a) no poner el acento.

b) escribir la palabra hasta la letra acentuada, añadir el código de la impresora correspondiente a BACKSPACE (vuelve un espacio hacia atrás) e imprimir el apóstrofo. Por ejemplo: para imprimir la palabra 'número', deberás escribir «nu» & chr$(8) & «mero». O algo más elegante: al principio del programa poner B$=CHR$(8) & CHR$(96), y entonces escribes «nu» & b$ & «mero».

Procedimientos y funciones

Una de las mejores cosas que tiene el QL es la definición de palabras para facilitar la programación. Cuando el ordenador encuentra, al correr el programa, una de estas palabras, busca la definición y ejecuta todas las instrucciones que se hallan entre las sentencias DEFINE (la palabra) y END DEFINE. Ahora voy a comentar

```
100 MODE 1
110 TRA 1: REMark (Quitar esta linea si no es la versiòn espa!ola)
120 OPEN #5, ser1
130 DIM unid$(9,6), dieci$(10,10), decen$(9,9), cent$(9,13)
140 RESTORE : datos
150 CLS: CSIZE 2,1
160 AT 3,0: INPUT "] Nùmero de loteria ?"!lt$
170 lot$="": FOR n=1 TO LEN(lt$): lot$=lot$ & lt$(n) & " "
180 AT 5,0: INPUT "] Pesetas ?"!num$: IF num$="" OR num$ < "1" OR num$ > "1000":
 CLS 3: GO TO 180
190 CSIZE 0,0
200 num=num$: traduce num
210 coloca sal$
220 CLS
230 avisa
240 imprime
250 IF otro: GO TO 150
260 CLS: STOP
270 :
280 :
290 DATA "UNA","DIEZ","CERO","CIENTO","DOS","ONCE","VEINTE","DOSCIENTAS"
300 DATA "TRES","DOCE","TREINTA","TRESCIENTAS","CUATRO","TRECE","CUARENTA","CUAT
ROCIENTAS"
310 DATA "CINCO","CATORCE","CINCUENTA","QUINIENTAS","SEIS","QUINCE","SESENTA","S
EISCIENTAS"
320 DATA "SIETE","DIECISEIS","SETENTA","SETECIENTAS","OCHO","DIECISIETE","OCHENT
A","OCHOCIENTAS"
```

APLICACION

qué hacen los procedimientos y la única función definidos en este programa:

— «datos» (Líneas 620-650): Carga los matrices unid$, dieci$, decen$ y cent$ con sus respectivos valores.

— «traduce» (670-1070): Este procedimiento es el más interesante del programa. Convierte la cantidad de pesetas que hemos introducido en número a su correspondiente literal, y lo asigna a la variable SAL$.

— «imprime» (1090-1180): lee los DATA de las líneas 340 a 590 y los envía a la impresora. En estas líneas está el recibo propiamente dicho, y naturalmente, si se cambian, cambiará también el resultado. Así que ya sabes, si quieres hacer otro modelo diferente, sólo tienes que cambiar estos DATA.

— «coloca» (1200-1270): devuelve la variable SAL$ más o menos centrada entre dos filas de '$'.

— «avisa» (1290-1360): se ase-

```
330 DATA "NUEVE","DIECIOCHO","NOVENTA","NOVECIENTAS","DIECINUEVE"
340 DATA "  ************************************************************  "
350 DATA " *                                                            * "
360 DATA "*                  L O T E R I.A   N A C I O N A L             *"
370 DATA "*                  ================================            *"
380 DATA "*                                                              *"
390 DATA "*          ^       NUMERO:        " & CHR$(14) & lot$ & CHR$(20) & "
     *"
400 DATA "*         ^^^                                                  *"
410 DATA "*        ^^^^^                                                 *"
420 DATA "*       ^^^^^^^                      El  portador  de  este    *"
430 DATA "*         ^^^              recibo  juega  la  cantidad  de     *"
440 DATA "*       ^^^^^^^            " & sal$ & "        *"
450 DATA "*      ^^^^^^^^^           pesetas  en  el  nùmero  arriba     *"
460 DATA "*         ^^^              indicado para el sorteo que  se     *"
470 DATA "*      ^^^^^^^^^           celebrarà  en  Madrid,  el  dia     *"
480 DATA "*     ^^^^^^^^^^^          21  de  Diciembre  de  1.985.       *"
490 DATA "*         ^^^                                                  *"
500 DATA "*         ^^^                      El  depositario,            *"
510 DATA "*         ^^^                                                  *"
520 DATA "*         ^^^                                                  *"
530 DATA "* FELIZ NAVIDAD                                                *"
540 DATA "*                                                              *"
550 DATA "*                                                              *"
560 DATA "*    SON: #" & num$ & "# PTS." & FILL$(" ",4-LEN(num$)) & "
                              *"
570 DATA "*    " & FILL$("=";12+LEN(num$)) & FILL$(" ",4-LEN(num$)) & "
                              *"
580 DATA " *                                                            * "
590 DATA "  ************************************************************  "
600 :
610 :
620 DEFine PROCedure datos
630 FOR n=1 TO 9: READ unid$(n), dieci$(n), decen$(n), cent$(n)
640 n=n+1: READ dieci$(n)
650 END DEFine
660 :
670 DEFine PROCedure traduce (a)
680 REMark LOCal nu,nu1,m,m1
690 nu1=a: sal$=""
700 SELect ON a
710  ON a=1
720   sal$=" UNA ": RETurn
730  ON a=1000
740   sal$=" MIL ": RETurn
750 END SELect
760 m=INT (nu1/100): m1=nu1-(100*m)
770 SELect ON nu1
780  ON nu1=100
790   sal$=sal$ & " CIEN ": RETurn
800  ON nu1=0 TO 99
810   GO TO 1000
820 END SELect
830 sal$=sal$ & " " & cent$(m)
840 SELect ON m1
850  ON m1=10 TO 19
860   sal$=sal$ & " " & dieci$(m1-9) & " ": RETurn
870  ON m1=0
880   sal$=sal$ & " ": RETurn
890  ON m1=0 TO 9
900   sal$=sal$ & " " & unid$(m1) & " ": RETurn
```

gura de que la impresora esté conectada cuando la envía lo códigos del espaciado (línea 1350).

— «otros» (1380-1500): esta función devuelve '1' si queremos hacer otra participación o '0' si no queremos.

Javier Ortega

UN EJEMPLO DE LO QUE HACE:

```
*****************************************************************
*                  L O T E R I A   N A C I O N A L              *
*                  ===============================              *
*        ^         NUMERO:          Ø  Ø  Ø  Ø  Ø               *
*       ^^^                                                     *
*      ^^^^^                         El  portador  de  este     *
*     ^^^^^^^            recibo  juega  la  cantidad  de        *
*       ^^^              $$$$$$$$$$$$$ CIEN $$$$$$$$$$$$        *
*     ^^^^^^^            pesetas  en  el  nùmero  arriba        *
*    ^^^^^^^^^           indicado para el sorteo que se         *
*       ^^^              celebrarà  en  Madrid,  el  dìa        *
*   ^^^^^^^^^^           21  de  Diciembre  de  1.985.          *
*  ^^^^^^^^^^^^                                                 *
*       ^^^                    El  depositario,                 *
*       ^^^                                                     *
*       ^^^                                                     *
*       ^^^                                                     *
* FELIZ NAVIDAD                                                 *
*                                                               *
*   SON: #1ØØ# PTS.                                             *
*   ===============                                             *
*****************************************************************
```

```
910 END SELect
920 nu=INT (m1/10): m1=m1-(10*nu)
930 IF NOT m1
940   sal$=sal$ & " " & decen$(nu) & " ": RETurn
950 END IF
960 IF nu=2
970   sal$=sal$ & " VEINTI" & unid$(m1) & " ": RETurn
980 END IF
990 sal$=sal$ & " " & decen$(nu) & " Y " & unid$(m1) & " ": RETurn
1000 SELect ON nu1
1010  ON nu1=10 TO 19
1020   sal$=sal$ & " " & dieci$(nu1-9) & " ": RETurn
1030  ON nu1=0 TO 9
1040   sal$=sal$ & " " & unid$(nu1) & " ": RETurn
1050 END SELect
1060 GO TO 920
1070 END DEFine
1080 :
1090 DEFine PROCedure imprime
1100 LOCal a$
1110 RESTORE 340
1120 REPeat impre
1130 IF EOF: EXIT impre
1140 READ a$
1150 PRINT# 5; a$
1160 END REPeat impre
1170 FOR n=1 TO 6: PRINT #5
1180 END DEFine
1190 :
1200 DEFine PROCedure coloca (a$)
1210 LOCal a1$,a2$,lon,mit
1220 lon=LEN(sal$): mit=INT(lon/2)
1230 a1$=sal$(1 TO mit): IF LEN(a1$)<16: a1$=FILL$("$",16-LEN(a1$)) & a1$
1240 a2$=sal$(mit +1 TO lon): IF LEN(a2$)<15: a2$=a2$ & FILL$("$",15-LEN(a2$))
1250 a$=a1$ & a2$
1260 RETurn a$
1270 END DEFine
1280 :
1290 DEFine PROCedure avisa
1300 CLS
1310 CSIZE 1,1
1320 AT 4,18: PRINT "Prepara la impresora"\TO 17;"y pulsa cualquier tecla"
1330 CSIZE 0,0
1340 PAUSE
1350 CLS: PRINT #5, CHR$(27);"0": REMark .....(espaciado a 1/8")......
1360 RETurn
1370 :
1380 DEFine FuNction otro
1390 LOCal a
1400 CLS
1410 CSIZE 2,1
1420 AT 4,7: PRINT "] Màs recibos ?  (S/N)"
1430 CSIZE 0,0
1440 REPeat lazo
1450   g$=INKEY$
1460   IF g$="s" OR g$="S": a=1: EXIT lazo
1470   IF g$="n" OR g$="N": a=0: EXIT lazo
1480 END REPeat lazo
1490 RETurn a
1500 END DEFine
1510 :
```

APLICACION

ARCHIVO DE ARCHIVES

Como muchos conoceréis, el Archive del QL es un potente programa que nos permite, mediante comandos, tener acceso fácil y rápido a una base de datos. Pero tiene el inconveniente de tener que teclear todo el comando, produciendo errores al no escribirlo correctamente y, además, los usuarios están acostumbrados al sistema de Menú para acceder a los datos sin tener que recordar que hay que abrir los ficheros.

Con el siguiente programa se pueden realizar las operaciones clásicas de altas, bajas, modificaciones, consultas o listados fácilmente.

Para ello se carga el programa Archive y mediante edit se teclea el listado que ofrecemos a continuación. Al ejecutarlo, pregunta por el fichero a inspeccionar. Tecleando "?" y ENTER le dará un catálogo de los ficheros existentes.

Después se plantea un menú de funciones, pudiéndose acceder al resto de las funciones de Archive pulsando la tecla "C". Para entrar de nuevo en el programa se teclea "Maestro".

Vicente Galán

```
proc confirmar
  escribir en 15,12;"confirme (s/n)?";
  haz yes=mayús(tecla())="S"
  escribir rept(" ",80)
  finproc
proc fin
  escribir en 15,0;"ABANDONAR ?";
  confirmar
  si yes
    limpiar
    cerrar
    escribir "ADIOS"
    stop
    sino
    cerrar
    start
    finsi
  finproc
proc maestros
  haz l$="z"
  escribir en 13,0;"P=Primero: U=Ultimo: S=Siguiente: A=Anterior (CONSULTAS PANT
ALLA).          "
  escribir "I=Insertar: B=Baja: M=Modificar: L=Listados impresora: F=Fin: C=Coma
ndo.          "
  escribir rept(" ",80)
  mientras l$<>"C"
    pescribir
    haz l$=mayús(tecla())
    si l$="P": primero : finsi
    si l$="U": último : finsi
    si l$="S": próximo : finsi
    si l$="A": anterior : finsi
    si l$="I": insertar : finsi
    si l$="B": escribir en 15,0;"BAJA";:confirmar: si yes: borrar : finsi : fins
i
    si l$="M": escribir en 15,0;"MODIFICAR ?";:CONFIRMAR: si yes: alterar : fins
i : finsi
    si l$="L": escribir en 15,0;"IMPRESORA";:confirmar: si yes: volcar : finsi :
 finsi
    si l$="F": escribir en 15,0;"FIN";:confirmar: si yes: volcar : finsi : finsi

    finmientras
  local t$
  escribir en 13,0;"elegir, reset, buscar, hallar, continuar, escribir cuenta(),
 escribir memoria(), imprimir,"
  escribir "ordenar, crear, situar, exportar, actualizar, haz, si, todos, otros
procedimientos"
  escribir en 15,0;rept(" ",80);"Pulse una tecla para continuar"
  haz t$=tecla()
  escribir en 15,0;"Acuerdese de volver al procedimiento con 'maestro'."
  finproc
proc start
  limpiar
  haz r$="?"
  escribir tab 15;"PROCEDIMIENTO MAESTRO; @ Vicente Galan Garcia"
  mientras r$="?"
    leer en 14,15;"Nombre del fichero?                ?=catálogo     ";r$
    si r$="?": dir "mdv2_": finsi
    finmientras
  abrir r$
  indicar
  maestros
  finproc
```

GRAFICOS EN

Indudablemente el QL es una máquina excepcional en muchos aspectos, sobre todo teniendo en cuenta su precio, ya que muchas de sus características sólo se encuentran en máquinas que cuestan mucho más. Por desgracia muchas de estas características son desconocidas para la mayoría de la gente que desaprovecha el gran potencial que ofrece la máquina.

Un ejemplo de éstos son los gráficos. Actualmente muy pocos programas saben sacar partido a esta característica, pero esto se puede paliar construyéndonos nuestros propios programas. En este artículo vamos a mostrar algunos de los posibles manejos que se pueden hacer desde BASIC y esperamos que usted se anime y haga más pruebas que nos muestren su imaginación

Empezando

Todos los programas que damos en este artículo se realizan con el modo normal de pantalla según se pone al encenderla, no obstante, si la ha modificado debera restituirla a su estado normal mediante los siguientes comandos:

FILL 0:OVER 1:MODE 8: SCALE 100,0,0

En este modo la pantalla del QL se encuentra dividida en pequeñas casillas formando un cuadriculado de 101 casillas verticalmente y 166 horizontalmente. Estas están numeradas de 0 a 100 y de 0 a 165, respectivamente, estando situados los orígenes de ambas en la parte interior izquierda. Cada uno de estos puntos puede ser direccionado indicando sus dos coordenadas, de modo que (45,89) indica el punto situado en la columna 45 y en la línea 89. Una vez establecido esto podemos

EL SUPERBASIC

aprender la primera instrucción gráfica, POINT X,Y. Donde X e Y son las coordenadas horizontal y vertical de un punto tal como indicamos antes. Lo que hace esta instrucción es poner la posición indicada del color que tuviésemos fijado previamente con INK. El programa de la figura uno utiliza esta instrucción para generar un anillo multicolor en la pantalla. En el listado se ve que existe una variable llamadas DIST que es la que mide la distancia al centro, de modo que no se dibuje fuera de los bordes fijados. Dado que este programa realiza un bucle infinito, se tiene que parar por medio de las teclas CTRL y espacio a la vez.

Otra instrucción de gran utilidad es LINE A,B TO C,D. Donde A,B y C,D son dos pares de puntos. Lo que hace el ordenador al recibir esta instrucción, como se habrá imaginado ya, es trazar una línea entre ambos puntos con el color fijado previamente. Aunque es una instrucción muy sencilla, con ella se pueden conseguir efectos espectaculares, como el que da el listado de la figura 2, en la que se han unido los puntos de dos líneas perpendiculares consiguiendo un efecto parecido a una red.

Otro ejemplo del uso de estas instrucción se da en el listado de la figura 3, que realiza una figura conocida como cardioide si el número que se le da es 2, para otros valores superiores genera otras curvas también de gran belleza y conocidas genéricamente con el nombre de epicicloides.

Curvas

Hasta ahora todas las líneas que hemos usado eran rectas, pero nuestra habilidad puede aumentar considerablemente si usamos curvas. La más sencilla de estas formas geométricas es el círculo, y existe una instrucción especial que nos permite trazarlo. Esta instrucción es CIRCLE X,Y,R. Donde X e Y constituyen las coordenadas del punto del centro (según el formato indicado anteriormente) y R es el radio. Como en ocasiones anteriores, podemos generar una figura exclusivamente a base de círculos pero que sin embargo posea gran belleza. El listado de un ejemplo de esto se da en el listado 4, que realiza una serie de círculos concéntricos con un radio que decrece aleatoriamente.

Otra instrucción existente en el QL y que no es normal en otros ordenadores es la ELLIPSE X,Y,R,EX,ANG. Todos aquellos que posean fundamentos de matemáticas gráficas sabrán que el círculo es en realidad un caso particular de la elipse, o dicho de otro modo, que una elipse es un circulo «de goma» que se estira por uno de sus lados. Para hacer esto en nuestro ordenador debemos fijar algunos parámetros. Los tres primeros ya los conocemos ya que son los mismos que en el círculo, y los dos últimos son los que fijan la excentricidad (achatamiento si se prefiere). Esto se fija en realidad con el primero, que cuando vale 1 hace un círculo y si es menor hace que esté estirado hacia arriba y si es mayor que 1, hace que estirado hacia los lados. El último parámetro, que hemos denominado ángulo, hace que la elipse gire en la pantalla con el ángulo indicado (en radianes), de modo que (para los que estén interesados en el efecto matemático) resulta lo mismo un factor de excentricidad de 0,5 y un ángulo de 0 que un factor de 2 y un ángulo de pi/4 radianes.

En el listado 5 se muestra un ejemplo de aplicación de este comando para dibujar secciones de un toro (un donuts para los amigos).

Con estas instrucciones ya se pueden realizar auténticas maravillas gráficas, aunque nos faltan algunos comandos más que se explicarán con más profundidad en el próximo mes.

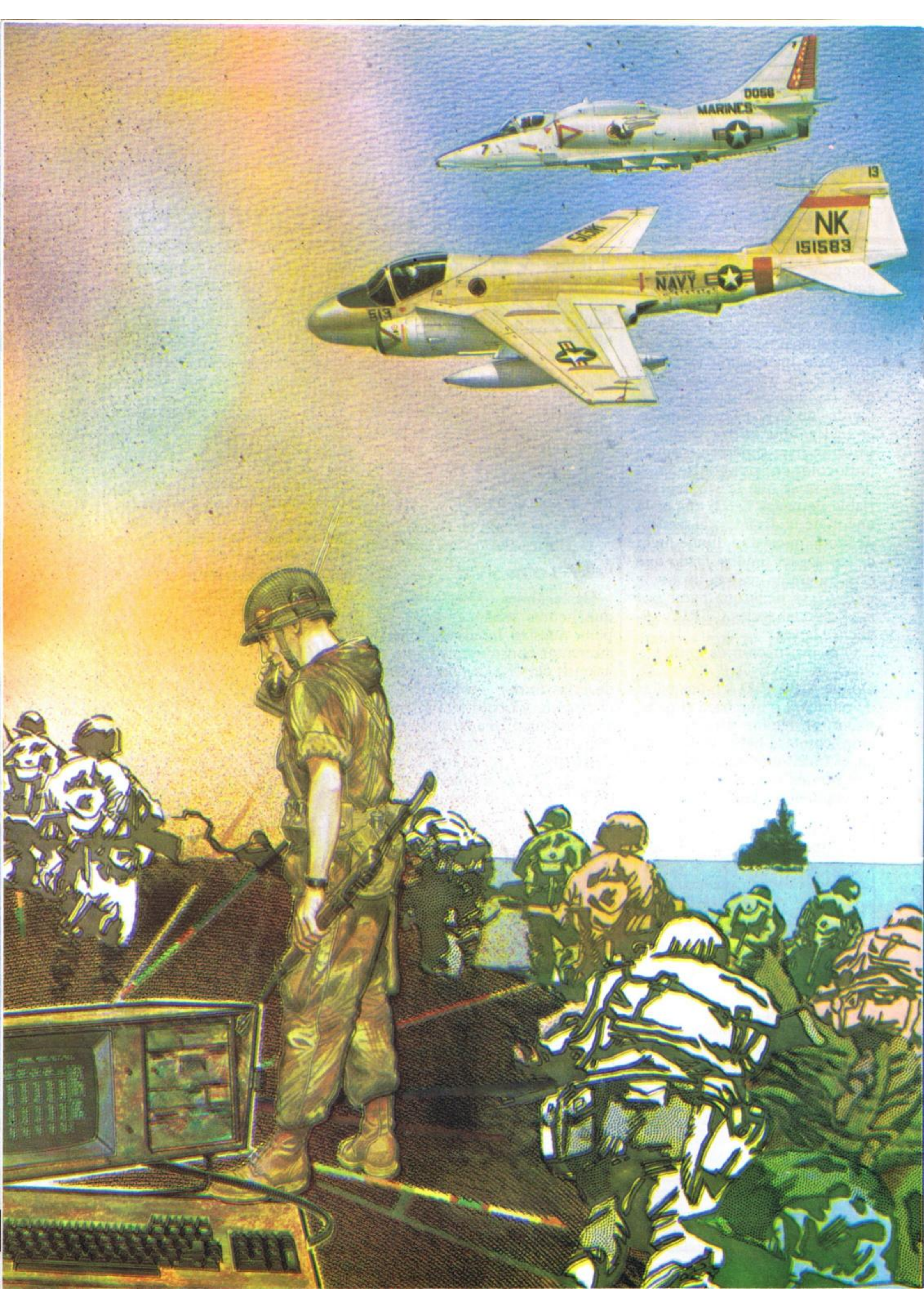
```
100 PAPER 0:CLS
110 REPeat loop
120 x=RND(20,80):y=RND(20,80)
130 dist=(x-50)*(x-50)+(y-50)*(y-50)
140 IF dist<900 AND dist>350 THEN
150 INK dist/100-2
160 POINT x,y
170 END IF
180 END REPeat loop
```

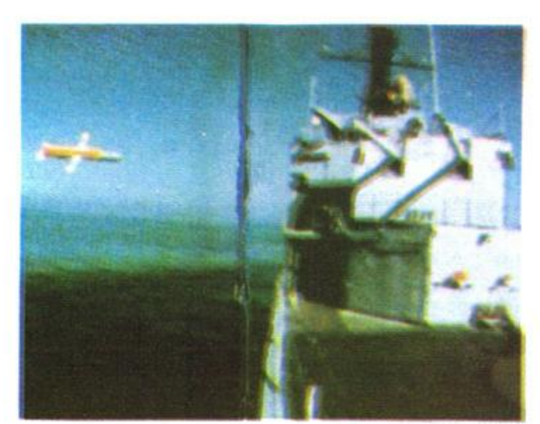
```
90 CLS
100 FOR i=0 TO 100 STEP 5
110 LINE i,0 TO 0,100-i
120 NEXT i
```

```
100 PAPER 1:INK 7:CLS
110 INPUT#0,"Multiplo a usar"!k
120 FOR angle=0 TO 360 STEP 3
130 a=RAD(angle)
140 x1=80+50*COS(a)
150 y1=50+50*SIN(a)
160 x2=80+50*COS(k*a)
170 y2=50+50*SIN(k*a)
180 LINE x1,y1 TO x2,y2
190 END FOR angle
```

```
100 PAPER 1:INK 7:CLS
110 FOR a=0 TO 360 STEP 10
120 t=RAD(a)
130 x=80+50*COS(t)
140 y=50+20*SIN(t)
150 r=20-y/8
160 ex=COS(t)*COS(t)
170 ELLIPSE x,y,r,ex,0
180 END FOR a
```

```
70 viejo=50
80 x=50
90 lado=1
100 CLS
110 radio=50
120 REPeat dibujo
130 IF lado=1 THEN x=x-viejo+radio:lado=0: ELSE x=x+viejo-radio:lado=1
140 CIRCLE x,50,radio
150 viejo=radio
160 radio=radio-RND(1 TO 4)
170 IF radio<6 THEN EXIT dibujo
180 END REPeat dibujo
```


MARINES
NK
151583
NAVY
513



La última

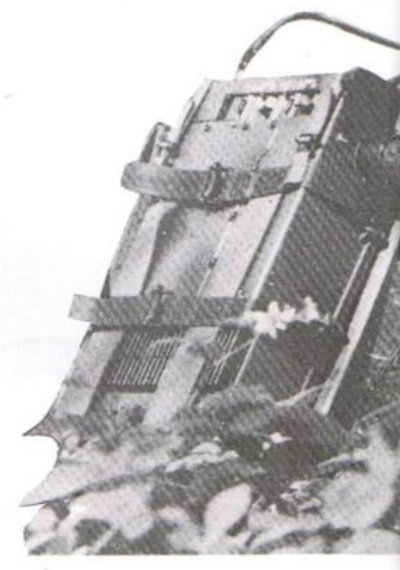




batalla



Este interesante programa que nos ha enviado Angel Soria constituirá una verdadera delicia para todos aquéllos que disfruten haciendo simulaciones de batallas en casa. El programa es fundamentalmente estratégico y por tanto no tiene una estructura de matar marcianos sino un planteamiento más «serio», ya que cada jugador (dos como mínimo) debe distribuir sus ejércitos entre todos sus países y atacar en función del equilibrio de fuerzas existentes. Este planteamiento es similar al juego del tablero de Risk que muchos conocerán y las reglas son similares, aunque las repetimos más adelante para aquéllos que las desconozcan.



La estructura interna es algo complicada, ya que hay cuatro programas distintos. El primero (listado 1) se encarga de cargar la pantalla de presentación y el programa principal (listado 3), éste a su vez, antes de ponerse en marcha carga un pequeño bloque de tipo CODE en el que hallan diversos datos, incluyendo los UDG, el mapa de pantalla y la calavera que aparece al final del juego.

Este bloque lo crea el listado 4 y la pantalla de presentación con el listado 2. Por tanto, para hacer una cinta que funcione, hay que grabar primero el programa 1 con SAVE «nombre» LINE 1, a continuación se graba la pantalla de presentación al ejecutar el programa 2, después se escribe el tercer listado, también con SAVE «nombre» LINE 1 y por último el resultado del programa cuatro. Un consejo importante: guarde en cinta aparte todos los programas antes de ejecutarlos. Un código máquina equivocado puede destruir el programa.

Si se desea suprimir la pantalla de presentación y el cargador inicial, hay que introducir la línea

```
4 RANDOMIZE: CLEAR 52299
```

en el programa principal.

Si al ejecutar el programa cuatro da error en las sentencias DATA,



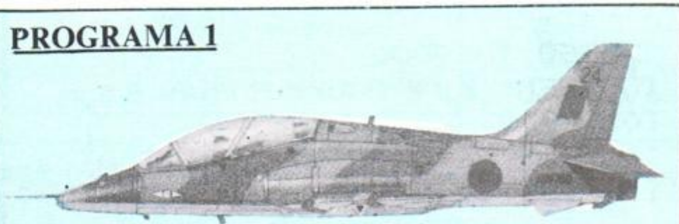
indicará la línea donde está éste, por lo que basta con revisarla. Si el error fuese que el mapa no aparece bien definido, habría que revisar de la 10 a la 250, estas mismas serían las responsables de que los continentes no se rellenen correctamente.

El programa principal

La estructura del programa principal es compleja debido a su longitud y a que era necesario para dotarlo de mayor rapidez. No obstante los bloques principales son los que siguen, indicados por líneas:

1-25 Carga de bytes y prepara algunas matrices.
100- Subrutinas.
110- Borra parte inf. pantalla.
120- borra desde la línea 5.
130- espera la pulsación de una tecla.
140-: Dibuja pantalla.
150 y 170-: Músicas.
180-: Búsqueda de propiedades de evaluación del objetivo.
190-: Busca si hay más de un territorio con 1 ejército, con el fin de conceder uno si no lo hay.
500-: Efectos para cuando se consigue el objetivo.
1000-: Subrutina para la colocación de ejércitos.
1500-: Efectos para cuando un jugador pierde sus territorios.
2000-: Programa principal, con el menú principal.
2035-: Salto a las opciones.
2040-2185-: Se comprueba si un jugador ha conseguido su objetivo. Sólo se comprueba el del jugador de turno.
2200-2210-: Siguiente jugador y salto a menú.
2290: Submenú.
2500: Rutina de atacar.

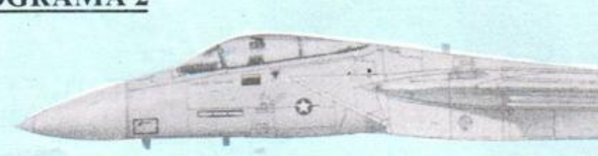
PROGRAMA 1



```
  10 CLEAR 52299
  20 BEEP .1,30: INK 0: BORDER 0
: PAPER 0: CLS
  30 PRINT AT 10,8; PAPER 5; INK
 1;"THE LAST BATTLE"; FLASH 1;AT
 12,10;"is loading."
  40 LOAD ""SCREEN$
  50 PRINT AT 20,0;
  60 LOAD ""
```



PROGRAMA 2



```
  10 BRIGHT 1: BORDER 0: PAPER 0
: INK 0: FLASH 1: CLS
  20 FOR m=5 TO 12: FOR n=0 TO 3
1: READ a
  30 PRINT  PAPER 6*(a>1); INK 2
*((a=1) OR (a=3));AT m,n;"▓"
  40 NEXT n: NEXT m
  50 PRINT  INK 7; FLASH 0;AT 3,
0;"  * * *  W A R  G A M E  * *
*  "
  60 PRINT  INK 7; FLASH 0;AT 14
,0;"  * * *  W A R  G A M E  * *
 *  "
  70 INK 5: FOR n=15 TO 21: PRIN
T AT n,0; FLASH 0,,: NEXT n
  80 PLOT 180,10: DRAW 20,28: DR
AW -10,-32: DRAW -10,14,1: DRAW
40,0
  85 PLOT 206,24: DRAW 5,6,3: DR
AW 0,6,-4: PLOT 201,24: PLOT 216
,24
  90 PRINT  FLASH 0;AT 17,23; IN
K 2;";AT 20,25;"1985"
  95 PRINT AT 18,8; INK 4;"LOADI
NG"
 100 DATA 0,1,1,1,0,1,0,1,0,1,1,
1,0,0,0,0,1,0,0,0,1,1,1,0,1,1,1,
0,1,1,1,0
 105 DATA 0,0,1,0,0,1,0,1,0,1,0,
0,0,0,0,0,1,0,0,0,1,0,1,0,1,0,0,
0,0,1,0,0
 110 DATA 0,0,1,0,0,1,1,1,0,1,1,
0,0,0,0,0,1,0,0,0,1,1,1,0,1,1,1,
```

3500-2550: Pide los datos.
2555-2707: Efectúa el combate.
2710-2800: Mov. Tropas.
3000: Rutina de información.
3500: Rutina de ver el objetivo.
4000: Efectos para fin desde menú.
8500: Subrutina de control del mapa.
9000: Instrucciones y toma de datos, así como inicialización.
9500: Datas para las matrices.

Variables:
nj = n.º jugadores.
tj = tipo juego 1 = C. mundo.
a$ (nj, 8) = Nombres de los jugadores.
j = jugador de turno.
o() = objetivo.
E() = todo sobre los territorios.
z() = empleada en el control del mapa. (POKES).
p$() = nombre países.
pa y pd = país ataca/defiende.
r = refuerzo. (Ejércitos de reserva).
d() = puntos dados.
m() = medallas.
j() =jugador que ha perdido todo.

Código máquina:
En él están el mapa, la calavera y los gráficos (UDG), además de esta rutina:

```
LD DE,4000h (16384)
LD HL,E458h (58456)
LD BC,1800h (6885)
LDIR
RET
```

Los valores de DE y HL se cambian para la calavera (también BC).

Instrucciones de uso

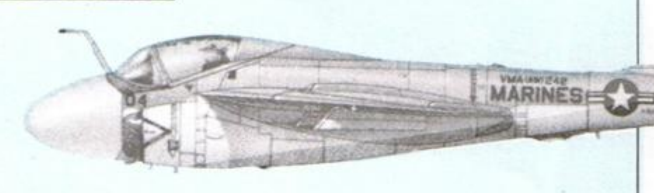
El juego es para varias personas (de 2 a 4) no pudiéndose jugar contra el ordenador.

Existen dos modalidades de juego, en la primera se lucha para conseguir el objetivo específico y la segunda es para conquistar el mundo. Avisamos que en este segundo caso la partida puede durar varios días, ya que en el primero dura varias horas.

PROGRAMA 2

```
0,0,1,0,0
 115 DATA 0,0,1,0,0,1,2,3,0,1,2,
2,2,0,2,2,3,0,2,2,3,0,3,0,0,0,3,
2,2,1,0,0
 120 DATA 0,0,1,0,0,1,2,1,2,1,3,
1,2,0,0,2,1,1,1,2,1,0,3,0,1,1,3,
0,0,1,0,0
 125 DATA 0,0,0,0,0,0,2,2,0,0,2,
2,2,0,0,2,0,0,0,2,0,0,2,0,0,0,2,
2,0,0,0,0
 130 DATA 0,0,0,0,0,0,2,0,2,0,2,
0,2,0,0,2,0,0,0,2,0,0,2,0,0,0,2,
0,0,0,0,0
 135 DATA 0,0,0,0,0,0,2,2,0,0,2,
0,2,0,0,2,0,0,0,2,0,0,2,2,2,0,2,
2,2,0,0,0
 140 SAVE "TLBsp"SCREEN$
```

PROGRAMA 3




```
 1 REM
 2 REM ▓▓▓ A.S. 1985 ▓▓▓
 3 REM
 4 RANDOMIZE
 5 PRINT AT 20,0;: LOAD ""CODE

 10 PRINT AT 18,4; INK 4; PAPER
 0; FLASH 1;"STOP THE TAPE": FOR
n=1 TO 10: BEEP .01,25: BEEP .0
1,20: NEXT n
 15 DIM z(40): RESTORE 9550: FO
R n=1 TO 40: READ a: LET z(n)=a:
 NEXT n
 20 RESTORE 9530: DIM p$(40,13)
: FOR n=1 TO 40: READ a$: LET p$
(n)=a$+".              ": NEXT n
 25 GO TO 9000
 100 REM ▓▓▓ Subrrutinas ▓▓▓
 105 REM
 110 FOR a=18 TO 21: PRINT AT a,
1;"                              "
: NEXT a: RETURN
 120 FOR m=5 TO 20: BEEP .01,m:
PRINT AT m,0,,: NEXT m: RETURN
 130 PRINT AT 21,7; INK 2; FLASH
 1;">"; FLASH 0; INK 1;"Pulsa un
a tecla.": PAUSE 0: RETURN
 140 CLS : PRINT  INK 1;AT 1,6;"
* THE LAST BATTLE *"; PAPER 0;AT
 3,0,,
 145 PRINT AT 0,0;"▛";AT 0,31;"▜
";AT 2,0;"▙";AT 2,31;"▟";AT 4,0;
"▛";AT 4,31;"▜";AT 21,0;"▙";AT 2
1,31;"▟": RETURN
 150 RESTORE 160: FOR m=1 TO 19:
 READ a: BEEP .15,a: NEXT m
 155 FOR m=1 TO 2: RESTORE 165:
FOR n=1 TO 16: READ a: BEEP .14,
a: NEXT n: NEXT m: RETURN
 160 DATA 2,4,5,2,5,5,4,2,4,-3,4
,5,7,4,7,7,5,4,2,5,4,2
 165 DATA 9,14,12,14,12,10,10,9,
```

Guía del comprador de Todospectrum

- **ALSISTOCKS**: Acceso directo (2 segundos, 1.800 artículos por fichero y cartucho, 14.400 en disco 800 K.
- **ALSIMAIL**: 10 ficheros con impresión de recibos mensuales, mailing, acceso directo, 800 fichas por fichero y cartucho.
- **CAMBIALSI**: Impresión de letras de cambio y recibos negociables.
- **ALSICONT**: 8.000 asientos en cartucho microdrive, 64.000 en disco 800 K, 1,3 segundos por asiento, balances y extractos inmediatos, 2 niveles, subcuentas en todas las cuentas, cantidad de cuentas ilimitada, etc.
- **COMERCIAL 6**: Facturación, almacén, ficheros, pedidos, presupuestos, estadísticas, relaciones, mailing, albaranes, etc.
- **ALSIFIN**: Todo tipo de cálculos financieros.
- **ALSIFINCAS**: Administración de fincas.

Oferta: Sinclair QL + impresora + monitor + ud. discos + lote programas 242.750 Ptas.

Oferta: Sinclair QL + impresora + lote programas 149.750 Ptas.

ALSI comercial, S. A. Antonio López, 117, 2.° D - 28026 MADRID - Telf. 475 43 39



DISTRIBUIDORES DE:
- COMMODORE-64
- ORIC-ATMOS
- ZX SPECTRUM
- SINCLAIR ZX 81
- ROCKWELL'-AIM-65
- DRAGON-32
- NEW BRAIN
- DRAGON-64
- CASIO FP-200

ELECTRONICA SANDOVAL, S. A.
C/ SANDOVAL, 3, 4, 6. 28010-MADRID
Teléfonos: 445 75 58 - 445 76 00 - 445 18 70
447 42 01
C/ SANDOVAL, 4 y 6
Centralita 445 18 33 (8 líneas)

ORDENADORES
- QL - AMSTRAD - SPECTRUM

PROGRAMAS
- Contabilidad QL .. **20.000** ptas.
- Nóminas QL ...... **25.000** ptas.



World-Micro s.a.
Avda. del Mediterráneo, 7
Tels. 251 12 00 y 251 12 09 - MADRID 7



Cada uno	Caja de 10	Caja de 30
C-5 199 ptas.	1.393 ptas.	3.582 ptas.
C-10 209 ptas.	1.463 ptas.	3.762 ptas.
C-15 219 ptas.	1.533 ptas.	3.942 ptas.
C-20 229 ptas.	1.602 ptas.	4.122 ptas.

Libre de gastos de envío contra reembolso correos

CAMAFEO INC. Dep. 03
José Lázaro Galdiano, 1. 28036 Madrid.



PROTEJA SU SPECTRUM PLUS CON ESTA PRACTICA FUNDA

A UN PRECIO ESPECIAL

OFERTA LIMITADA Y EXCLUSIVA PARA NUESTROS LECTORES



Aproveche la oportunidad de mantener como nuevo su Spectrum Plus con esta funda, y beneficiese de un 30% de descuento sobre su precio normal.

¡APRESURESE! RECORTE Y ENVIE HOY MISMO ESTE CUPON A:
PUBLINFORMATICA (Dto. FUNDAS), C/BRAVO MURILLO, 377 5.° A 28020 MADRID

CUPON DE PEDIDO

Sí, envíeme al precio de 975 Ptas. cada una, ____ fundas para mi SPECTRUM PLUS
El importe lo abonaré: Con mi tarjeta de crédito □ American Express □
Visa □ Interbank □ Adjunto cheque □
Contra reembolso □
Número de mi tarjeta ____
Fecha de caducidad ____
NOMBRE ____
DIRECCION ____
CIUDAD ____
C.P. ____ PROVINCIA ____
Sin gastos de envío



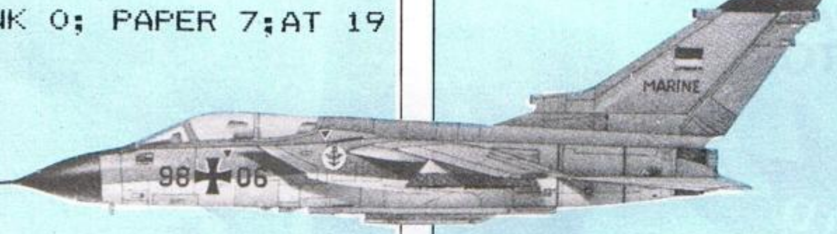
Una vez puesto en marcha el programa se nos pide el número de jugadores y a continuación el modo de juego, siendo «c» la conquista del mundo y «o» la de un objetivo específico.

A continuación se introducen los nombres de los jugadores (hasta 8 caracteres) y luego empieza el juego, eligiendo el ordenador aleatoriamente quien empieza primero.

Si se ha elegido la conquista de un objetivo se pide entonces que se introduzca la clave secreta que nos permitirá ver cuál es nuestro blanco. Esta clave, evidentemente, es distinta para cada jugador y tiene

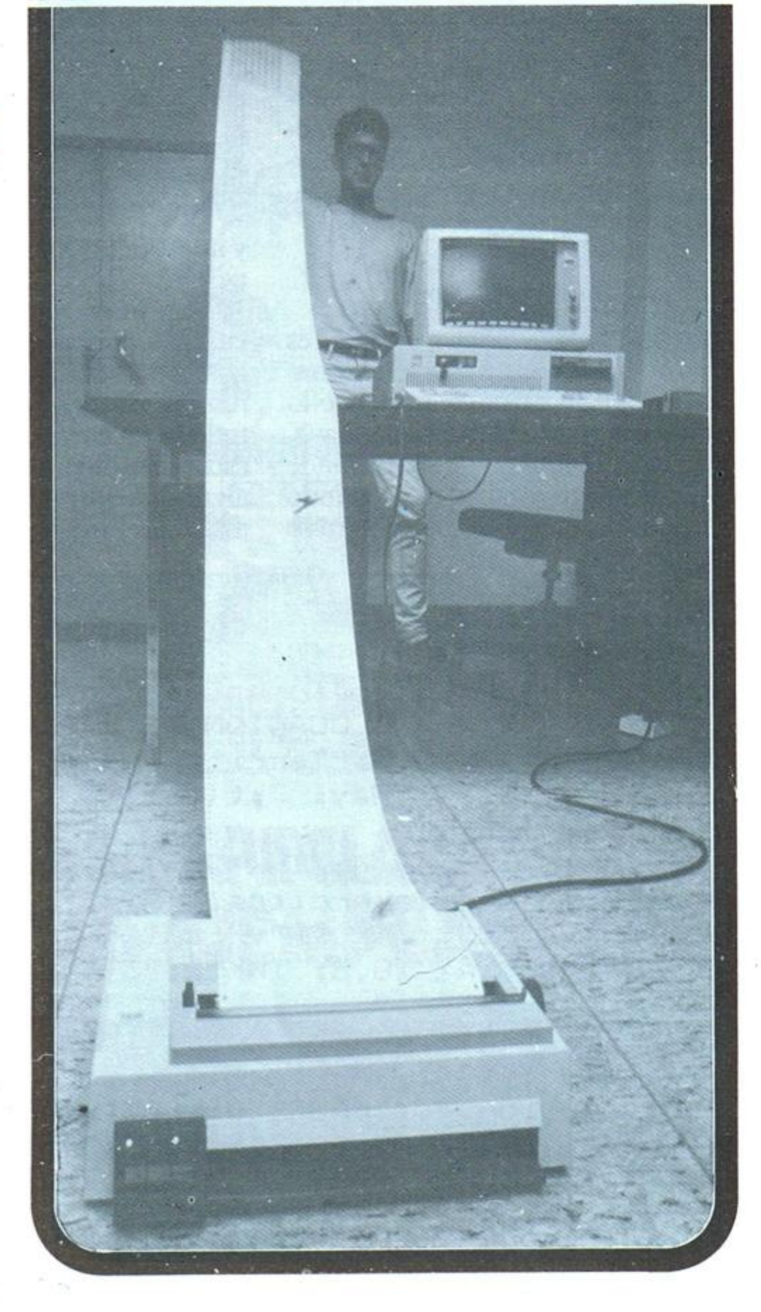
```
7,2,10,7,9,5,4,-2,5,4,2
 170 RESTORE 175: FOR m=1 TO 12:
 READ a: BEEP .15,a: NEXT m: RET
URN
 175 DATA 7,7,12,12,11,11,5,5,9,
9,7,7
 180 FOR n=a TO c: IF t(n,1)=j2
THEN  LET b=b+1
 185 NEXT n: RETURN
 190 LET r1=0: FOR m=1 TO 40: IF
 t(m,1)=j1 THEN  IF t(m,2)>1 THE
N  RETURN
 195 NEXT m: LET r1=1: RETURN
 500 REM  ▒▒▒ Final ▒▒▒
 505 REM
 510 PAPER 0: INK 7: CLS
 515 LET oo=USR 58444: BEEP .5,3
0:
 520 PRINT  INK 0; PAPER 7;AT 19
,6;"* THE LAST BATTLE *"
 525 POKE 58448,76: POKE 58449,2
04: POKE 58452,24
 530 FOR m=1 TO 2: FOR n=0 TO 7
STEP .5: BEEP .05,-10: BEEP .05,
-5: BORDER n: NEXT n: NEXT m
 535 BORDER 0
 540 FOR n=1 TO 20 STEP 2: INK I
NT (n/2.6)
 545 BEEP .04,10
```

```
 550 CLS : LET oo=USR 58444
 555 PAUSE n*.7
 560 NEXT n
 565 POKE 58448,88: POKE 58449,2
28: POKE 58452,27
 570 FOR n=1 TO 156
 575 PRINT AT 21,0; PAPER 6; INK
 2;("
     LA ULTIMA BATALLA HA TENIDO
 LUGAR         "+a$(j2)+" ha con
seguido su objetivo.           L
A GUERRA HA DESTRUIDO EL MUNDO
         R.I.P.
        ")(n TO 31+n)
 580 BEEP .05,30: NEXT n
 590 BEEP 2,-10
 600 FOR m=14 TO 10 STEP -1: FOR
 n=.1 TO .002 STEP -.01: BEEP n,
m: BEEP n,m-5: NEXT n: NEXT m
 610 PAUSE 5: BEEP 1,-10: PAUSE
50: GO TO 9000
1000 REM  ▒▒▒ Poner Ejerc. ▒▒▒
1005 REM
1020 LET oo=USR 58444: GO SUB 85
00
1025 IF t(t,1)<>j1 THEN  BEEP .1
,0: PRINT AT 21,7; INK 2; FLASH
1;"No te pertenece.": PAUSE 50:
GO SUB 110: GO SUB 8550: GO TO 1
```



CONCURSO MATEMATICO

El número más largo



Este impresionante listado de doce páginas es el mayor número primo conocido. El ordenador (un potente IBM AT) tardó seis horas y media en generarlo usando un programa compilado.

Dado que el mes de diciembre ofrece grandes ratos de descanso a la mayoría de nuestros lectores, la redacción de la revista ha decidido proporcionar a todos nuestros lectores una oportunidad de aprovechar esas horas a la vez que mejoran sus dotes programadoras por medio de este concurso.

El objetivo a conseguir es descomponer un número no primo en sus divisores, cosa no muy difícil normalmente, pero en este caso algo más complicada, ya que el número en cuestión tiene 29 cifras y, por tanto, no se puede calcular normalmente recurriendo a trucos como los que aparecen en el artículo de infinita precisión que aparece en este mismo número.

Esperamos que os animéis a participar y nos demostréis vuestro alto nivel.

Bases del concurso

1. Los concursantes deberán descomponer el número 23479159600157356175629850677 en sus factores primos (que no sean a su vez divisibles por ningún otro), indicando cuántas veces aparece cada uno. Este cálculo puede realizarse en una ordenador Sinclair (ZX80, ZX81, ZX Spectrum, ZX Spectrum+, ZX Spectrum 128 o QL).

2. Todos los participantes deberán enviar al concurso: dichos factores primos junto con la cantidad de veces que aparecen. El programa empleado (indicando el lenguaje e incluyendo una cinta o microdrivre con el listado) y el tiempo aproximado que tardó en resolver el problema. Si este tiempo fuese pequeño (lo dudamos) deberán indicarse también los segundos, en el caso normal basta con decir las horas y los minutos aproximados. Estos tiempos y el buen funcionamiento del programa estarán sujetos a comprobación.

3. Podrán participar todos los lectores de la revista que no trabajen en la redacción de la misma.

4. El jurado estará compuesto por la redacción de Todospectrum.

5. A la hora de puntuar se tendrá en cuenta: la rapidez (considerando el lenguaje empleado), el buen diseño del programa y, naturalmente, que los resultados sean correctos.

6. La fecha tope de admisión de programas es el día 15 de enero. La relación de premiados aparecerá en el número de febrero.

7. Cada concursante podrá participar cuantas veces quiera, pero en cada una de ellas debe hacerlo con un programa distinto (aunque es conveniente que los resultados coincidan).

8. Al ganador se le entregará un lápiz digitalizador de la marca PIN SOFT, una colección de Anaya Multimedia y una colección de juegos ABC Soft.

Entre los demás participantes se sortearán otros nuevo lápices digitalizadores de PIN SOFT.

Recuerda: Hay que hallar todos los divisores (con el número de veces que aparece cada uno) del número:
23479159600157356175629850677.

que tener un máximo de tres caracteres.

Después de esto el ordenador informa del número de territorios de cada jugador y del número de ejércitos disponibles (3,5 por el número de territorios) y después de darnos información acerca de las teclas de manejo empieza el juego.

Colocación de ejércitos

Lo primero que hay que hacer es repartir los ejércitos que el ordenador ha dejado sin colocar, para ello hay que mover el cursor con Z y X y colocarlo encima del país donde queramos quitar o poner, cuando estemos en él pulsamos el espacio



y aparece un menú con tres opciones:

1. Poner ejércitos. Nos pide cuántos ponemos (de 0 a 9). Si se colocan todos los que quedan en reserva pueden pasar dos cosas, si todo se ha correctamente se sale de esta parte de programa, pero si hay algún territorio sin ejércitos nos anulará el movimiento, ya que no puede haber ningún territorio sin ejércitos. Tampoco se pueden dejar ejércitos en reserva, por lo que no se saldrá de esta opción hasta no colocarlos todos.

2. Quitar ejércitos. Funciona de un modo similar a la anterior pero preguntando los ejércitos que se quitan y que van a la reserva.

3. Esta opción nos devuelve al menú principal.

Aunque de una sola operación se pueden quitar o poner un máximo de nueve ejércitos de cada vez, esta operación se puede repetir y meter más ejércitos, no existiendo límite al número de ejércitos que puede tener un país.

PROGRAMA 3

```
025
1030 GO SUB 140
1035 PRINT AT 8,7;"Turno de ";a$
(j1);AT 6,2;"- COLOCACION DE EJE
RCITOS -";AT 10,5;"Territorio: "
;p$(t);AT 11,5;"Hay: ";t(t,2);"
ejercitos."
1040 PRINT AT 16,6; INK 2;"1"; I
NK 1;".- Poner ejercitos.";AT 17
,6; INK 2;"2"; INK 1;".- Quitar
ejercitos.";AT 18,6; INK 2;"3";
INK 1;".- Salir."
1042 PRINT AT 20,4;"        "; INK
2; FLASH 1;">"; INK 1; FLASH 0;"
Pulsa opcion.",: PAUSE 0: LET x$
=INKEY$: BEEP .05,20: IF x$<"1"
OR x$>"3" THEN  BEEP .1,0: GO TO
 1040
1043 IF x$="3" THEN  GO TO 1000
1045 LET y$="quitas": IF x$="1"
THEN  LET y$="pones": GO TO 1050
1050 PRINT AT 16,0,,,,,,: IF x$=
"1" THEN  PRINT AT 16,6;"Tienes
";(STR$ r+" ")( TO 2);" ejercito
s."
1055 PRINT AT 20,4;"Cuantos ";y$
;" (0...9)? ";: PAUSE 0: LET z$=
INKEY$: BEEP .05,20: PRINT z$: I
F z$<"0" OR z$>"9" THEN  BEEP .1
,0: GO TO 1055
1057 IF x$="2" THEN  IF (VAL z$>
=t(t,2)) OR z$="0" THEN  PRINT A
T 20,4;" No puedes hacer eso.",:
 PAUSE 50: GO TO 1040
1060 IF x$="2" THEN  LET t(t,2)=
t(t,2)-VAL z$: LET r=r+VAL z$: B
EEP .1,20: PRINT AT 11,10;t(t,2)
;" ejercitos.  ";AT 16,6;"Tienes
 ";r;" ejercitos.": PAUSE 80: GO
 TO 1080
1062 IF x$="1" THEN  IF r-VAL z$
<0 THEN  PRINT AT 20,4;" No pued
es hacer eso.",: PAUSE 50: GO TO
 1040
1065 LET b=0: FOR m=1 TO 40: IF
t(m,1)=j1 THEN  IF t(m,2)=0 THEN
  LEJ b=b+1: IF t=m THEN  IF z$<
>"0" THEN  LET b=b-1
1070 NEXT m: IF b<>0 AND r-VAL z
$<b THEN  BEEP .1,0: PRINT AT 19
,1;"Los necesitas para ocupar lo
s  territorios  desocupados,",A
T 21,5; PAPER 5;"Movimiento anul
ado.": PAUSE 100: GO TO 1000
1075 BEEP .05,10: LET t(t,2)=t(t
,2)+VAL z$: LET r=r-VAL z$: PRIN
T AT 11,10;t(t,2);" ejercitos.
";AT 16,13;(STR$ r+"  ")( TO 2):
```

SERVICIO DE EJEMPLARES ATRASADOS

Complete su colección de

Todospectrum

A continuación le resumimos el contenido de los ejemplares aparecidos hasta ahora.

Núm. 1 • 250 pts.
Cómo usar el microdrive/Programación Basic/Ampliación Basicare/Rutina despertador/Variables del sistema/Entrada datos mediante máscaras/Protección del software/Sintonice su Spectrum/Programas.

Núm. 2 • 250 pts.
Gráficos profesionales/Desplazamiento pixel a pixel/Utilización de rutinas/Construcción del interface centronics/Programas de utilidad para microdrive/Rutina reset en código máquina/Análisis del editor de textos Tasword/Interfaces para impresoras/Programas.

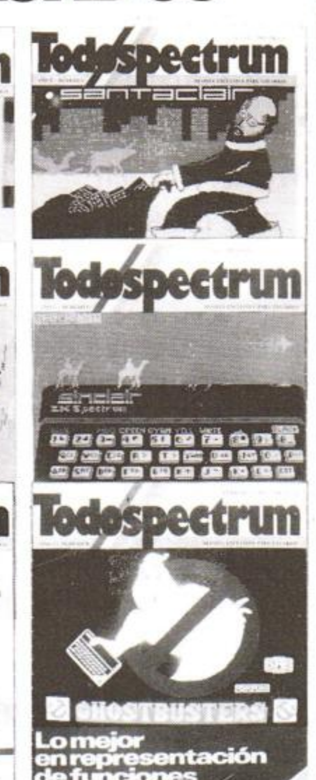
Núm. 3 • 250 pts.
Novedades sonimag '84/Ampliando el Basic/Programas para ordenar programas/Gráficos con el VU-3D/Lenguaje Forth/Archivos en microdrive/Programación de un interface de impresora/Programas.

Núm. 4 • 250 pts.
De profesión: programador/Consola para el Spectrum/Comparación código máquina-Basic/Análisis programa contabilidad /Calendario/Pascal/Programas.

Núm. 5 • 250 pts.
Floppys para Spectrum/Diseño asistido por ordenador/64 Caracteres por línea/Juego de la vida/Pascal/Así hacemos las portadas/Control de evaluaciones/Programas.

Núm. 6 • 250 pts.
Representación de funciones/Todos los caminos conducen a la ROM/Juegos/Pascal/Construcción de un lápiz óptico/Programas de gestión. El SITI/Logo: tortugas para todos/Interrupciones del Z-80/Programas.

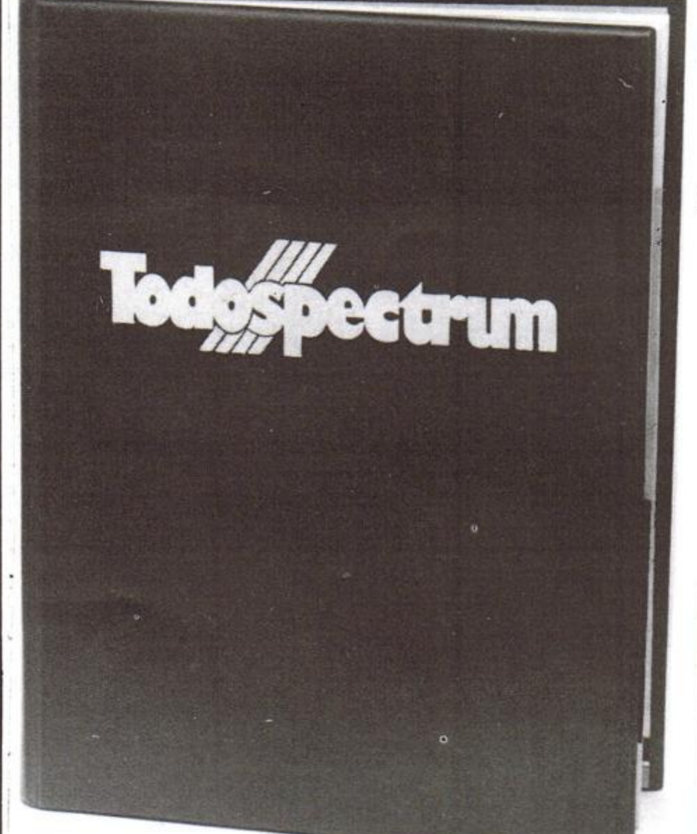




DISPONEMOS DE TAPAS ESPECIALES PARA SUS EJEMPLARES DE

Todospectrum

SIN NECESIDAD DE ENCUADERNACION



(cada tapa es para 6 ejemplares)



Para hacer su pedido, rellene este cupón HOY MISMO y envíelo a:

Todospectrum **Bravo Murillo, 377**
Tel. 733 96 62 - 28020 MADRID

Ruego me envíen los siguientes ejemplares atrasados de TODOSPECTRUM .... al precio de 250 pts.

Por favor envíenme .... tapas para la encuadernación de mis ejemplares de TODOSPECTRUM, al precio de 600 pts. más gastos de envío.

El importe lo abonaré
□ POR CHEQUE □ CONTRA REEMBOLSO □ CON MI TARJETA DE CREDITO □ AMERICAN EXPRESS □ VISA □ INTERBANK

Número de mi tarjeta: □□□□□□□□□□□□□□□□□□□□

Fecha de caducidad .... Firma

NOMBRE ....

DIRECCION ....

CIUDAD .... C. P. ....

PROVINCIA ....

Cuando todos los jugadores finalizan de colocar sus ejércitos se pasa al juego propiamente dicho.

Menú principal

Este también ofrece varias opciones que detallamos a continuación.

1. Atacar. Esta opción nos muestra el mapa del mundo y se nos pide que elijamos el país que ataca (que debe ser nuestro) y el que defiende (que debe ser del contrario), entre ambos debe haber frontera, es decir, deben estar el uno al lado del otro. Una vez elegidos ambos se les pide al atacante que diga con cuántos ejércitos ataca (hasta un máximo de tres) y luego al defensor de le pregunta lo mismo.



El número de ejércitos con los que se ataca o defiende es el número de dados que tira ese jugador (en realidad lo hace el ordenador) y sólo se considera el dado más alto. La puntuación más alta de los dos indica el que gana y en caso de empate gana el defensor.

Por ejemplo el atacante va con tres ejércitos, y saca 1 3 3 y si el defensor tiene dos ejércitos y saca 2 y 4, gana el defensor por tener un cuatro mientras que el atacante ha obtenido solamente un tres como máximo.

El jugador que pierda se quedará sin los ejércitos con los que ataca o defiende. Si un jugador pierde todos los ejércitos de un territorio, éste pasa a poder del contrincante.

Medallas:

1. Si gana el atacante, aparecerán las medallas que posee y se le concede una (no aparece en la lis-

```
 PAUSE 80
1080 IF r=0 THEN  RETURN
1090 GO TO 1000
1500 REM ▮▮▮Fin de jugador▮▮▮
1505 REM
1510 POKE 58448,76: POKE 58449,2
04: POKE 58452,24
1515 BEEP 1.5,-40
1520 PAPER 0: INK 7: CLS : LET o
o=USR 58444
1525 FOR f=1 TO 84: PAUSE 3: PRI
NT  PAPER 6; INK 2; BRIGHT 1;AT
21,0;("
       Para el  jugador  "+a$(m)+"
 ha sido su ultima batalla.
         R.I.P.
 ")(f TO f+31): BEEP .01,35: NEX
T f
1530 POKE 58448,88: POKE 58449,2
28: POKE 58452,27
1540 FOR f=1 TO 8: FOR z=40 TO 1
0 STEP -2: BEEP .01,z: NEXT z: N
EXT f
1545 INK 0: PAPER 7: RETURN
2000 REM  ▮▮▮ P.Principal  ▮▮▮
2005 REM
2010 BEEP .1,40: BEEP .01,30: GO
 SUB 140
2020 PRINT AT 6,7;"Turno de ";a$
(j);AT 9,8; INK 3;"MENU";AT 11,9
; INK 2;"1"; INK 1;".-Atacar."'T
AB 9; INK 2;"2"; INK 1;".-Inform
acion."'TAB 9; INK 2;"3"; INK 1;
".-Ver el Objetivo."'TAB 9; INK
2;"4"; INK 1;".-Fin del Juego."
2025 PRINT AT 20,1;"Introduzca o
pcion."; FLASH 1;"?": PAUSE 0: L
ET x$=INKEY$: BEEP .1,30
2030 IF x$<"1" OR x$>"4" THEN  B
EEP .1,0: GO TO 2025
2035 GO TO 2000+500*VAL x$
2040 GO SUB 140: IF tj=1 THEN  G
O TO 2150
2045 LET j2=j: LET b=0: GO TO 20
25+(o(j)*15 AND o(j)<8)+(125 AND
 o(j)>=8)+(15 AND o(j)=1)
2055 LET a=1: LET c=40: GO SUB 1
80
2060 IF b>=72/nj THEN  GO TO 500
2065 GO TO 2150
2070 LET a=10: LET c=13: GO SUB
180: LET a=26: LET c=35: GO SUB
180
2075 IF b=14 THEN  GO TO 500
2080 GO TO 2150
2085 LET a=1: LET c=9: GO SUB 18
0: LET a=14: LET c=19: GO SUB 18
0
2090 IF b=15 THEN  GO TO 500
2095 GO TO 2150
2100 LET a=10: LET c=13: GO SUB
180: LET a=20: LET c=25: GO SUB
180: LET a=36: LET c=40: GO SUB
180
2105 IF b=15 THEN  GO TO 500
2110 GO TO 2150
2115 LET a=1: LET c=9: GO SUB 18
0: LET a=14: LET c=19: GO SUB 18
```



PROGRAMA 3

```
0: LET a=36: LET c=40: GO SUB 18
0
2120 IF b=20 THEN  GO TO 500
2125 GO TO 2150
2130 LET a=20: LET c=35: GO SUB
180
2135 IF b=16 THEN  GO TO 500
2150 LET a=1: LET c=40: FOR m=1
TO nj: LET j2=m: LET b=0: GO SUB
 180
2155 IF b<>0 THEN  GO TO 2170
2160 IF o(j)=m+7 THEN  LET j2=z:
 GO TO 500
2165 IF j(m)<>1 THEN  LET j(m)=1
: GO SUB 1500
2170 NEXT m
2175 IF tj=2 THEN  GO TO 2200
2180 LET b=0: FOR n=1 TO nj: IF
j(n)=0 THEN  LET b=b+1: LET j2=n
2185 NEXT n: IF b=1 THEN  GO TO
500
2200 LET j=j+1: IF j>nj THEN  LE
T j=1
2205 IF j(j)=1 THEN  GO TO 2200
2210 BEEP .1,40: BEEP .01,30: GO
 TO 2020
2290 REM ▓▓▓Sub Menu. ▓▓▓
2295 REM
2400 GO SUB 140
2405 PRINT AT 6,7;"Turno de ";a$
(j);AT 9,8; INK 3;"MENU";AT 11,9
; INK 2;"1"; INK 1;".-Salir."'TA
B 9; INK 2;"2"; INK 1;".-Medalla
s."'TAB 9; INK 2;"3"; INK 1;".-M
ov. Tropas."
2410 PRINT AT 20,1;"Introduzca o
pcion."; FLASH 1;"?": PAUSE 0: L
ET x$=INKEY$: BEEP .1,30
2415 IF x$="1" THEN  GO TO 2040
2420 IF x$="2" THEN  LET ma=3: G
O SUB 140: GO TO 2715
2425 IF x$="3" THEN  GO SUB 110:
 GO TO 2805
2430 BEEP .1,0: GO TO 2410
2500 REM  ▓▓▓ Atacar ▓▓▓
2505 REM
2510 LET oo=USR 58444: FOR m=1 T
O 2
2515 IF m=1 THEN  PRINT AT 20,1;
">Atacante"; FLASH 1; INK 1;"?":
 GO TO 2525
2520 PRINT AT 20,1;"Defensor ";
FLASH 1; INK 1;"?"
2523 IF m=2 THEN  POKE b,(PEEK b
)-128
2525 GO SUB 8500+55*(m=2): BEEP
.1,30: PRINT AT 20,1;"
"
2530 IF (t(t,1)<>j AND m=1) OR (
t(t,1)=j AND m=2) THEN  BEEP .1,
0: PRINT AT 21,9; INK 2; FLASH 1
;"No se puede.": PAUSE 50: GO TO
 2510
2535 IF m=1 THEN  LET pa=t: GO T
O 2545
2540 LET pd=t: FOR d=3 TO 6: IF
t(pa,d)<>t THEN  NEXT d: PRINT
FLASH 1; INK 2;AT 21,7;"No hay f
rontera.": PAUSE 50: GO TO 2510
2545 BEEP .1,40: NEXT m
```

ta) y a veces un ejército. A la derecha de la tabla de medallas está la cotización en ejércitos de cada una, ya que se pueden canjear por éstos. El ordenador pregunta a continuación si se cambia, si se responde que sí, nos dará a elegir entre cinco opciones.

Si no se realiza el cambio se vuelve al movimiento de tropas y después al submenú.

2. Si gana el defensor, aparecen las medallas que éste posee, se le concede otra y un ejército, no habiendo en este caso opción para cambio de medallas ni de tropas.

El submenú tiene tres opciones distintas: la primera nos devuelve al menú principal, la segunda nos permite cambiar medallas por ejércitos y la tercera realiza los movimientos de tropas. A este menú se va independientemente de que se hayan realizado conquistas.

La segunda opción del menú principal nos da información de los ejércitos que posee cada jugador y de los territorios.

La tercera nos muestra el objetivo que tenemos que conquistar, previa introducción de la clave secreta.

La última opción del menú principal termina el juego.

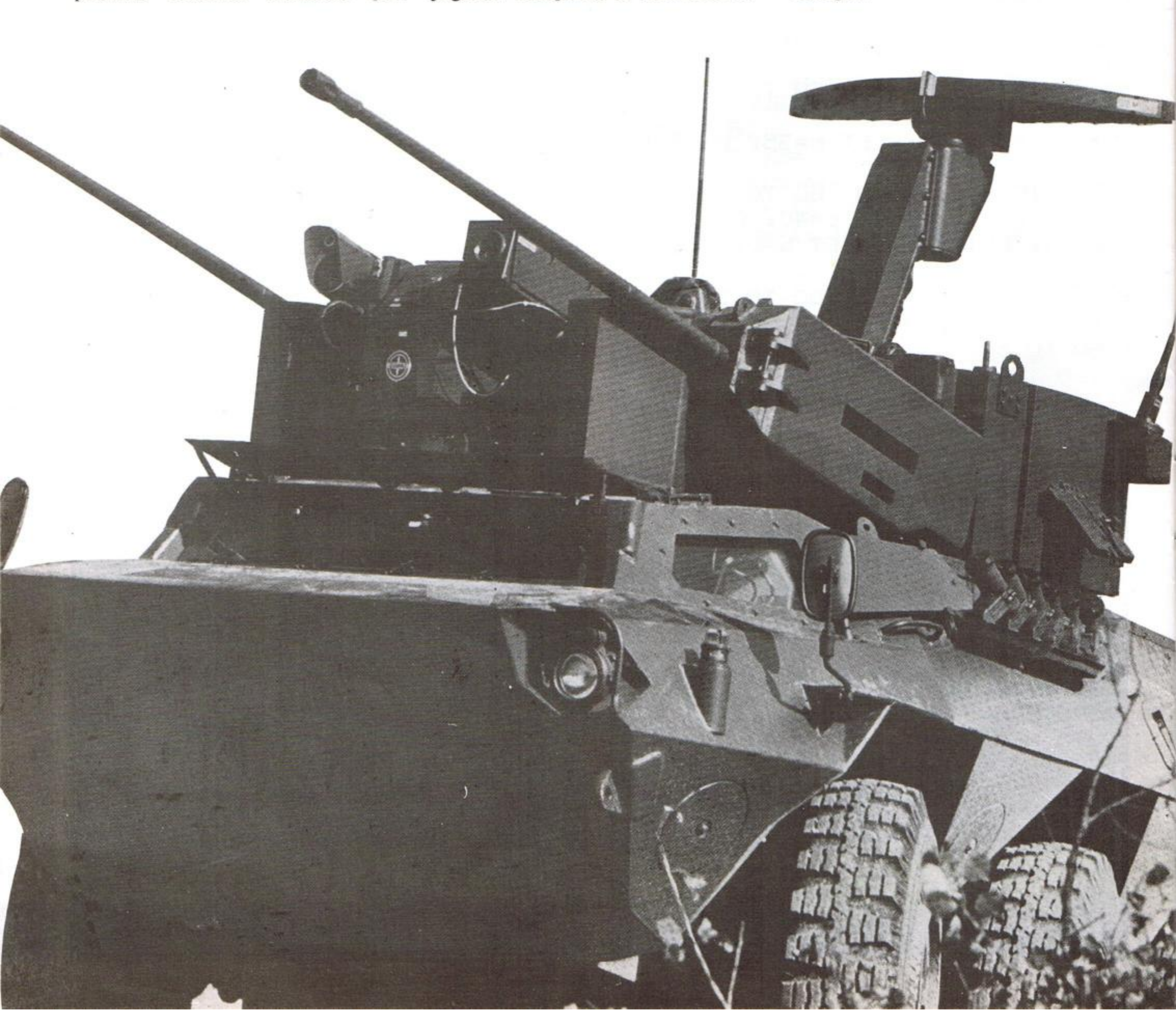
Resumen de los objetivos del juego

El juego, continúa hasta que un jugador conquista el territorio indicado, apareciendo entonces un dibujo y un mensaje. Si un jugador pierde todos sus territorios, queda eliminado, siendo esto indicado por el ordenador.

Si un jugador conquista un territorio y no posee ejércitos para ocuparlo, el ordenador le dará uno automáticamente.

El movimiento del cursor por el mapa va de continente en continente no siendo exactamente de derecha a izquierda o viceversa. Por ejemplo, si sale por Oceanía, se va a América.

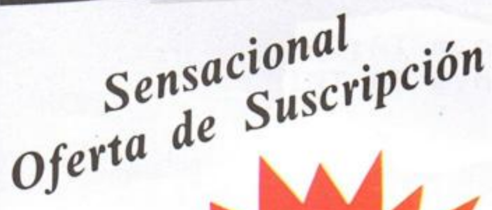
Con esto quedan explicadas todas las instrucciones del juego, ahora sólo queda teclear y saber estrategia.



Todospectrum



TODOSPECTRUM es una publicación mensual que le ayudará a obtener el máximo partido a su **SPECTRUM** y al **ZX 81**.



CONOZCA LAS VENTAJAS DE SUSCRIBIRSE A

Todospectrum





ADEMAS, le hacemos un 25 % DE DESCUENTO

sobre el precio real de suscripción (12 números)

VALOR REAL DE SUSCRIPCION	OFERTA ESPECIAL DE SUSCRIPCION	USTED AHORRA
~~3.600~~ PTAS.	2.700 PTAS.	900 PTAS.

APROVECHE AHORA esta oportunidad irrepetible para suscribirse a **TODOSPECTRUM**. Envie **HOY MISMO** la tarjeta adjunta a la revista, que no necesita sobre ni franqueo. Deposítela en el buzón más cercano. Inmediatamente recibirá su primer ejemplar de **TODOSPECTRUM** más el **REGALO**.

Todospectrum

Bravo Murillo, 377
Tel. 733 79 69
28020 MADRID



PROGRAMA 3

```
2550 GO SUB 140
2555 PRINT AT 6,2;">Ataca ";a$(j
)'TAB 4; INK 1;"Desde ";p$(pa)'T
AB 13;t(pa,2);" ejercitos."
2560 PRINT AT 11,2;">Defiende ";
a$(t(pd,1))'TAB 4; INK 1;"Desde
";p$(pd)'TAB 13;t(pd,2);" ejerci
tos."
2565 FOR n=1 TO 2: PRINT AT 20,3
;"Con cuantos ej. ";
2570 IF n=1 THEN  PRINT "atacas
"; FLASH 1;"?": GO TO 2580
2575 PRINT "defiendes "; FLASH 1
;"?"
2580 LET a=t(pd*(n=2)+pa*(n=1),2
): PAUSE 0: LET x$=INKEY$: BEEP
.01,30: IF x$<"1" OR x$>"3" THEN
 BEEP .1,0: GO TO 2580
2583 IF VAL x$>a THEN  BEEP .1,0
: GO TO 2580
2585 IF n=1 THEN  LET ea=VAL x$:
 PRINT  INK 1;AT 8,4;"Con ";ea;"
 de": GO TO 2595
2590 LET ed=VAL x$: PRINT AT 13,
4; INK 1;"Con ";ed;" de"
2595 NEXT n: PRINT AT 20,0,,
2600 FOR n=40 TO 176 STEP 136: F
OR m=0 TO 32 STEP 16: BEEP .005,
m: PLOT n+m,40: DRAW 7,0: DRAW 1
,-1: DRAW 0,-7: DRAW -1,-1: DRAW
 -7,0: DRAW -1,1: DRAW 0,7: NEXT
 m: NEXT n
2605 FOR n=2 TO 19 STEP 17: FOR
m=16 TO (ea AND n=2)+(ed AND n=1
9)+15
2610 PRINT AT m,n; INVERSE 1; IN
K 2;CHR$ (143+(t(pa,1) AND n=2)+
(t(pd,1) AND n=19))
2615 BEEP .01,10: NEXT m: NEXT n
2620 DIM d(2): FOR b=5 TO 22 STE
P 17: LET s=0: LET c=ea*(b=5)+ed
*(b=22): FOR n=1 TO c: FOR m=1 T
O 20: LET a=INT (RND*6)+1
2625 BRIGHT 1: IF c=3 THEN  PRIN
T AT 17,b;CHR$ (147+a)
2630 IF c>=2 AND n<3 THEN  PRINT
 AT 17,b+2;CHR$ (147+a)
2635 IF n=1 THEN  PRINT AT 17,b+
4;CHR$ (147+a)
2640 BEEP .01,-10: NEXT m: BEEP
```

```
.02,25: BEEP .02,35: BEEP .02,30
2645 IF a>s THEN  LET s=a
2650 NEXT n: PAUSE 10: BEEP .1,3
0
2655 LET d((b<>5)+1)=s: NEXT b:
BRIGHT 0
2660 IF d(1)<=d(2) THEN  LET t(p
a,2)=t(pa,2)-ea: PRINT AT 20,3;"
El atacante pierde ";ea;" ej.":
GO TO 2670
2665 LET t(pd,2)=t(pd,2)-ed: PRI
NT AT 20,3;"El defensor pierde "
;ed;" ej."
2670 PAUSE 30: LET n=19-17*(d(2)
>=d(1)): FOR m=16 TO ea*(d(2)>=d
(1))+ed*(d(2)<d(1))+15
2675 PRINT  OVER 1; INK 6;AT m,n
;"·": BEEP .05,-10: PRINT  OVER
1; INK 2;AT m,n;"*": BEEP .05,-1
2
2680 PRINT  INK 0;AT m,n;"░": BE
EP .05,-11: PRINT AT m,n;" ": BE
EP .05,-12
2685 NEXT m
2690 IF t(pd,2)<>0 AND t(pa,2)<>
0 THEN  LET j1=j: PAUSE 50: GO T
O 2400
2695 PAUSE 30: PRINT  PAPER 5;AT
 21,4;"Territorio conquistado"
2700 IF t(pd,2)=0 THEN  LET j1=j
: LET t(pd,1)=j1: LET ma=1: GO T
O 2707
2705 LET ma=0: LET j1=t(pd,1): L
ET t(pa,1)=j1
2707 GO SUB 170: PAUSE 50
2710 GO SUB 140
2715 PRINT  INK 3;AT 5,4;"- COND
ECORACIONES -"; INK 0;AT 7,4;"Co
ronel: ";a$(j1)
2717 PRINT AT 9,5;r;" ejercitos.
"
2720 FOR n=1 TO 3: FOR m=1 TO m(
j1,n): BEEP .01,n+m+10: PRINT AT
 9+n*2,m+8; INK n+1;CHR$ (158+n)
;AT 10+n*2,m+8;CHR$ (161+n): NEX
T m: NEXT n
2723 IF ma=3 THEN  GO TO 2740
2725 LET a=INT (RND*3)+1: LET m(
j1,a)=m(j1,a)+1
2727 GO SUB 190
2730 PRINT AT 19,5;"Se le conced
e una "; INK a+1;CHR$ (158+a);AT
 20,23;CHR$ (161+a)
2732 IF j1<>j THEN  LET r1=1
2735 IF RND>.7 OR r1=1 THEN  PRI
NT AT 20,6;"Y un ejercito.": LET
 r=1
2736 IF ma=0 THEN  GO TO 2810
2737 BEEP .1,20
2740 PRINT AT 7,22;"CAMBIO:";AT
9,22; INK 2;"▼▼▼";AT 10,22;"***"
; INK 0;"=5"
2745 PRINT  INK 3;AT 11,22;"┬┬┬"
;AT 12,22;"●●●"; INK 0;"=4"
2750 PRINT AT 13,22; INK 4;"▼▼▼"
;AT 14,22;"***"; INK 0;"=3"
2755 PRINT AT 15,22; INK 2;"▼▼";
 INK 4;"▼";AT 16,22; INK 2;"**";
 INK 4;"*"; INK 0;"=2"
2760 PRINT AT 17,22; INK 3;"┬┬";
 INK 4;"▼"; INK 3;AT 18,22;"●●";
 INK 4;"*"; INK 0;"=1"
2765 PRINT AT 21,10;"Cambias"; F
LASH 1; INK 2;"?": PAUSE 0: IF I
NKEY$="n" THEN  GO TO 2805
2770 BEEP .1,20: LET a=r: PRINT
AT 21,10;"Opcion ": PAUSE 0: LET
 x$=INKEY$: BEEP .1,30
2775 IF x$="5" THEN  IF m(j1,1)>
=3 THEN  LET r=r+5: LET m(j1,1)=
m(j1,1)-3
2780 IF x$="4" THEN  IF m(j1,2)>
=3 THEN  LET r=r+4: LET m(j1,2)=
m(j1,2)-3
2785 IF x$="3" THEN  IF m(j1,3)>
=3 THEN  LET r=r+3: LET m(j1,3)=
m(j1,3)-3
2790 IF x$="2" THEN  IF m(j1,1)>
=2 THEN  IF m(j1,3)>=1 THEN  LET
 r=r+2: LET m(j1,1)=m(j1,1)-2: L
ET m(j1,3)=m(j1,3)-1
2795 IF x$="1" THEN  IF m(j1,2)>
=2 THEN  IF m(j1,3)>=1 THEN  LET
 r=r+1: LET m(j1,2)=m(j1,2)-2: L
ET m(j1,3)=m(j1,3)-1
2800 LET ma=3: GO TO 2710
2805 BEEP .05,40: PRINT AT 21,5;
 FLASH 1; PAPER 6;">"; FLASH 0;"
Movimiento de tropas.": PAUSE 50
: GO SUB 1000: GO TO 2040
2810 BEEP .01,0: BEEP .01,30: LE
```



PROGRAMA 3

```
T t(pa,2)=1: LET j1=j: LET r=0:
GO SUB 130: GO TO 2400
3000 REM ▮▮▮ Informacion ▮▮▮
3005 REM
3010 GO SUB 120: PRINT  INK 4;AT
 6,7;"- INFORMACION -"
3015 LET s=0: LET c=0: FOR n=1 T
O nj: PRINT AT n+10,1; INK 2;n;
INK 1;".- Coronel: ";a$(n): NEXT
 n
3020 PRINT AT 20,1;"De quien qui
eres informacion?.": PAUSE 0: LE
T x$=INKEY$: BEEP .01,30: IF x$<
"1" OR x$>STR$ nj THEN  BEEP .1,
0: GO TO 3020
3025 GO SUB 120: LET b=VAL x$: L
ET a=7: PRINT  INK 4;AT 5,1;"-IN
FORMACION- >"; INK 0;a$(b): FOR
n=1 TO 40: IF t(n,1)<>b THEN  GO
 TO 3040
3030 IF a=19 THEN  GO SUB 130: G
O SUB 120: PRINT  INK 4;AT 5,1;"
-INFORMACION- >"; INK 0;a$(b): L
ET a=7
3035 LET a=a+1: LET s=s+t(n,2):
LET c=c+1: PRINT AT a,1; INK 2;"
>"; INK 0;p$(n),t(n,2);" ejercit
os."
3040 NEXT n: IF a>=16 THEN  GO S
UB 130: GO SUB 120: PRINT  INK 4
;AT 5,1;"-INFORMACION- >"; INK 0
;a$(b)
3045 PRINT '"     Total territ.:";
c'"   Total ejerc. :";s
3050 GO SUB 130: GO TO 2000
```

```
3500 REM ▓▓▓ Ver el obj ▓▓▓
3505 REM
3510 GO SUB 120: PRINT AT 5,1;"-
OBJETIVO- >";a$(j): IF tj=1 THEN
 PRINT AT 10,1;"Objetivo:"; INK
 4;" Conquistar el mundo.": GO S
UB 130: GO TO 2000
3515 INPUT " Introduzca su clave
 >"; LINE x$: LET x$=(x$+"   ")(
 TO 3)
3520 IF c$(j)<>x$ THEN  BEEP .1,
0; PRINT  PAPER 6;AT 20,7; FLASH
 1;"CLAVE INCORRECTA.": PAUSE 30
: GO TO 2000
3525 PRINT AT 10,1;"Objetivo:":
RESTORE 9500: FOR m=1 TO o(j): R
EAD x$: NEXT m: PRINT  INK 4;AT
12,1;x$
3530 GO SUB 130: GO TO 2000
4000 REM ▓▓▓ Fin desde menu ▓▓▓
4005 REM
4010 PRINT AT 20,0; FLASH 1;"
 Pulse O para confirmar.     ":
PAUSE 0: IF INKEY$<>"0" THEN  BE
EP .1,20: GO TO 2000
4015 BEEP 1,30: PAPER 7: INK 0:
CLS : PRINT AT 0,0;"▛";AT 0,31;"
▜";AT 21,0;"▙";AT 21,31;"▟"
4020 PLOT 75,130: DRAW 8,-2: DRA
W 16,-3,-2: DRAW 4,-3
4025 DRAW 70,35,-1.2: DRAW -37,-
50,1
4030 DRAW 32,-12,.5: DRAW -15,-2
5,-.5: DRAW -22,20,.5
4035 DRAW -5,-60,1: DRAW -25,82,
-1.2
4040 DRAW -27,18,-.5
4050 CIRCLE 88,128,1
4055 FOR n=1 TO 31: PRINT AT 20,
0; PAPER 4; BRIGHT 1;"
 HA TRIUNFADO              LA PAZ
                            "(n T
O n+31)
4060 BEEP .05,30: NEXT n
4065 FOR n=1 TO 10: FOR m=20 TO
40 STEP 3
4070 BEEP .01,m+n: BEEP .01,m-1+
n: NEXT m: NEXT n
4075 GO TO 9000
8500 REM ▓▓▓ Pantalla ▓▓▓
8505 REM
8515 RESTORE 9515: FOR n=1 TO 40
: READ a,b: PRINT  INK 8; PAPER
9;AT b,a;CHR$ (t(n,1)+143): NEXT
 n
8520 GO TO 8555
8525 LET b=z(t)+22000
8530 POKE b,(PEEK b)+128: PAUSE
0: LET x$=INKEY$: BEEP .007,25
8535 IF x$="z" THEN  LET t=t-1:
IF t=0 THEN  LET t=40
8540 IF x$="x" THEN  LET t=t+1:
IF t=41 THEN  LET t=1
8545 IF x$=" " THEN  BEEP .1,10:
 RETURN
8550 POKE b,(PEEK b-128)
8555 BEEP .005,30: PRINT AT 19,1
; INK 1;">"; INK 0;a$(t(t,1));AT
 19,15; INK 2;">"; INK 0;p$(t)
8560 PRINT AT 20,15; INK 2;">";
INK 0;t(t,2);" ejercitos."
8565 GO TO 8525
9000 REM ▓▓▓ Principio ▓▓▓
9005 REM
9010 POKE 23609,20
9015 INK 0: PAPER 7: BRIGHT 0: B
ORDER 0
9020 GO SUB 140
```



```
9025 PRINT AT 5,0; INK 1;" Juego
 de estrategia (War game)  en el
 que pueden participar de  2 a 4
 jugadores."'" Existen dos forma
s de juego:la  conquista del mun
do,o la de un  objetivo determin
ado, (Esta es  la mas corta)."
9030 PRINT  INK 1;" El jugador,p
rimero desplazara    sus ejercito
s, y despues lu-     chara, pudie
ndo luchar con 1,    2 o 3 ejerci
tos (es mas proba-   ble vencer c
on 3).(La lucha es   mediante los
 dados)."
9035 PRINT AT 20,2;"N. Jugadores
:"; FLASH 1;"?": PAUSE 0: LET z$
=INKEY$: BEEP .05,30: PRINT AT 2
0,15;z$: IF z$<"2" OR z$>"4" THE
N  BEEP .1,0: GO TO 9035
9040 LET nj=VAL z$
9045 PAUSE 50: PRINT AT 20,2;"C.
```

PROGRAMA 3

```
Mundo/Ojetivo:(c/o):"; FLASH 1;"
?": PAUSE 0: LET z$=INKEY$: BEEP
 .05,30: PRINT AT 20,24;z$: LET
tj=(1 AND z$="c")+(2 AND z$="o")
: IF tj=0 THEN  BEEP .1,0: GO TO
 9045
9050 PAUSE 20: GO SUB 120
9055 DIM j(nj)
9060 DIM a$(4,8): PRINT AT 5,0
9065 FOR n=1 TO nj: INPUT ("Nomb
re jug.";n;" ? "); LINE z$: IF z
$="" THEN  GO TO 9065
9070 LET z$=(z$+"        "): P
RINT '" Jugador ";n;" : ";z$( TO
 8); INK 1;" "; INVERSE 1;CHR$ (
143+n): LET a$(n)=z$
9075 NEXT n
9080 LET j=INT (RND*4)+1: IF j>n
j THEN  GO TO 9080
9085 DIM o(nj): PRINT '" OK. Com
ienza ";a$(j)
9087 IF tj=1 THEN  GO TO 9130
9090 PAUSE 50: GO SUB 120
9100 PRINT AT 6,2;"-Reparto de t
erritorios.-"'"     Asignacion de
 claves."
9105 PRINT  INK 2''" Para ver du
rante el juego el    objetivo de
beras introducir     tu clave."
9110 DIM c$(nj,3): FOR n=1 TO nj
: PRINT AT 20,2;"Coronel: ";a$(n
)
9115 INPUT "Elija codigo (xxx):
"; LINE x$: LET x$=x$+"   "( TO
3)
9120 LET c=INT (RND*11)+1: FOR m
=1 TO nj: IF o(m)=c THEN  GO TO
9120
9125 NEXT m: LET o(n)=c: LET c$(
n)=x$: IF o(n)>=7+n THEN  GO TO
9120
9127 RESTORE 9500: PRINT AT 17,1
;"Objetivo:": FOR m=1 TO o(n): R
EAD x$: NEXT m: PRINT  INK 4;" "
;x$: GO SUB 130: GO SUB 110: NEX
T n
9130 PAUSE 50: GO SUB 120: PRINT
 AT 6,2;"-Reparto de territorios
.-"
9135 PRINT  INK 1;AT 8,1;" Se da
 a cada jugador:"''"    - ";INT (
40/nj);" territorios."'"    - ";I
NT ((40/nj)*3.5);" ejercitos."
9140 IF nj=3 THEN  PRINT ''" Se
 sortea un terriritorio."
9145 DIM b(nj): DIM t(40,6): FOR
 n=1 TO 40
9150 LET a=INT (RND*nj)+1
9155 IF nj<>3 THEN  GO TO 9165
9160 IF b(1)=13 AND b(2)=13 AND
b(3)=13 THEN  GO TO 9170
9165 IF b(a)=INT (40/nj) THEN  G
O TO 9150
9170 LET b(a)=b(a)+1: LET t(n,1)
=a: NEXT n
9175 GO SUB 120
9180 PRINT  INK 2;AT 5,4;"-Tecla
s de juego.-"''"    Z -Cursor Izq
."'"    X -Cursor Dcha."'"    SPAC
E -Selecciona territorio."
9185 PRINT  INK 1''" El movimien
to del cursor no es exactamente
 este,ya que varia    con los ter
ritorios."
9190 PRINT ''" En los demas caso
s pulsar la     tecla que se indi
que."
9195 RESTORE 9560: FOR n=1 TO 40
: FOR m=3 TO 6: READ a: LET t(n,
m)=a: NEXT m: NEXT n
9200 GO SUB 120: PRINT AT 6,1;"-
Colocacion de ejercitos."
9205 DIM m(nj,3): PRINT  INK 3;A
T 9,0;" Los jugadores deben de c
olocar  sus ejercitos en sus pai
ses,     a condicion de que ningu
no     quede desocupado."'" El
ordenador coloca tres en      cad
a territorio, el resto los    col
```





```
oca el jugador."
9207 FOR n=1 TO 40: LET t(n,2)=3
: NEXT n: GO SUB 130
9210 LET t=1: FOR d=1 TO nj: LET
 r=INT ((40/nj)/2): LET j1=d: GO
 SUB 120: PRINT AT 6,7;"Turno de
 ";a$(d): GO SUB 130: GO SUB 100
0: NEXT d
9215 LET oo=USR 58444
9220 PRINT AT 19,6;"* THE LAST B
ATTLE *"
9225 FOR m=1 TO 10: FOR n=0 TO 7
: BEEP .01,n*2+m*2: BORDER n: NE
XT n: NEXT m
9230 BORDER 0: PAUSE 10: GO SUB
150: PAUSE 30: BEEP .1,40: GO TO
 2000
9500 REM  ▓▓▓ Datos ▓▓▓
9501 REM
9505 DATA "Conquistar "+STR$ INT
 (72/nj)+" territorios.","Conqui
star "+STR$ INT (72/nj)+" territ
orios.","Conquistar Asia y Am.Su
r.","Conquistar Am.Norte y Europ
a.","Conquistar Africa,Ocenania
y        America de Sur."
```

```
9510 DATA "Coquistar Am.Norte, E
uropa y       Oceania.","Conquist
ar Asia y Africa.","Destruir a "
+a$(1),"Destruir a "+a$(2),"Dest
ruir a "+a$(3),"Destruir a "+a$(
4)
9515 DATA 2,3,5,4,7,5,9,5,11,1,5
,6,7,7,6,8,7,9,9,10,9,12,11,12,9
,14
9520 DATA 14,2,13,4,13,6,15,5,16
,3,18,4,14,8,17,8,15,9,17,10,17,
12,20,12
9525 DATA 19,8,18,6,21,8,23,8,24
,5,27,6,23,3,27,3,20,4,20,6,22,1
1,25,12,27,13,28,10,29,15
9530 DATA "Alaska","Alberta","On
tario","Labrador","Groenlandia",
"California","Florida","Mejico",
"Centroamerica"
9535 DATA "Venezuela","Peru","Br
asil","Argentina","Islandia","In
glaterra","Europa del S","Aleman
ia","Escandinavia","Ucrania"
9540 DATA "Marruecos","Egipto","
Guinea","Etiopia","Sudafrica","M
agadascar","Arabia","Turquia","I
```

PROGRAMA 3

```
ndia","Indochina","China"
9545 DATA "Japon","Siberia","Kam
chatka","Ural","Afganistan","Ind
onesia","Autralia occ","Australi
a or","N.Ginea","N.Zelanda"
9550 DATA 626,661,695,697,571,72
5,759,790,823,857,921,923,985,60
6,669,733,703,640
9555 DATA 674,798,801,831,865,92
9,932,803,738,805,807,712,747,64
7,651,676,740,902,937,971,876,10
37
9560 DATA 2,33,0,0,1,5,6,3,2,6,4
,7,3,5,7,0,14,2,0,4,7,2,3,8,6,4,
8,3,6,7,9,0,8,10,0,0
9565 DATA 9,11,12,20,10,12,13,0,
10,11,13,22,11,12,0,0,5,15,18,0,
14,18,17,16,15,17,20,0,15,16,18,
19,14,15,17,19,17,18,34,27
9570 DATA 16,10,22,21,20,22,23,2
6,20,12,23,21,24,21,22,25,25,23,
0,0,24,23,0,0,21,27,28,35,19,34,
35,26,26,29,35,36,28,30,36,0
9575 DATA 31,32,29,0,30,33,0,0,3
0,34,33,0,32,31,1,0,19,35,27,32,
34,26,27,28,28,29,37,39,36,39,38
,0,37,39,40,0,37,38,36,0,38,0,0,
0
```

PROGRAMA 4

```
   1 REM
   2 REM  ▒▒▒ A.S. 1985  ▒▒▒
   3 REM
   5 GO SUB 2000: PRINT "Creando
 graficos.": GO SUB 3000
   7 PRINT '"Creando dibujos."'"
Tiempo aprox. 8m:40s": PAUSE 150
   8 CLEAR 52299: PAPER 5: BORDE
R 0: INK 0: FLASH 0: OVER 0: INV
ERSE 0: BRIGHT 0: CLS
  10 PLOT 72,50: DRAW 3,20: DRAW
 -4,6,2: DRAW -5,9,-1: DRAW 3,6,
-1
  15 DRAW -1,4,2: DRAW -11,7,-.5
: DRAW -7,5,-2: DRAW -3,6: DRAW
-2,0,3: DRAW 2,-6: DRAW -2,0,-3:
 DRAW -5,8
  20 DRAW -3,8,-1: DRAW -3,10,1:
 DRAW -10,6,2: DRAW -6,-5,2: DRA
W -3,2,-3: DRAW 0,8,-1: DRAW 1,5
,1: DRAW 7,4,-2
  25 DRAW 7,-1,-1: DRAW 8,-1,1:
DRAW 5,0,-2: DRAW 10,-1: DRAW 3,
1: DRAW 4,1,-1: DRAW 7,2,2
  30 DRAW 2,-3,-3: DRAW -8,-7,1:
 DRAW 0,-6,3: DRAW 4,-2,-2: DRAW
 5,0,3: DRAW 0,8,1
  35 DRAW 5,0,-2: DRAW 5,-4,1: D
RAW 5,-6,-1: DRAW 0,-4,-2: DRAW
-6,-3,-1
  40 DRAW -7,-3,1: DRAW -5,-7,-1
: DRAW 1,-5: DRAW -1,-1,-2: DRAW
 -3,4,1: DRAW -5,-10,3: DRAW 4,3
,2: DRAW 2,0,-2
  45 DRAW 0,-5: DRAW 3,-2,-1: DR
AW 5,-5,2: DRAW 10,-1,-2
  50 DRAW 12,-9,1: DRAW 3,-3,-2:
 DRAW -3,-10: DRAW -4,-4,-1: DRA
W -5,-8,-1: DRAW -5,-4,-1: DRAW
-2,-5,-1: DRAW -1,-5,-1
  55 DRAW 2,-2: DRAW -2,-2,-3: D
RAW -6,6,-2
  60 PLOT 235,151: DRAW -15,3,1:
 DRAW -8,1,-1: DRAW -5,3,1: DRAW
```



```
-4,-1,2: DRAW -8,1,-2: DRAW -8,
0,1: DRAW 0,3,-2: DRAW 0,3,2: DR
AW -9,-1,1
  65 DRAW -9,-7,1: DRAW -2,0,-3:
 DRAW -3,0,3: DRAW 0,-2,-2: DRAW
 -8,-1,1: DRAW -8,-2,2
  70 DRAW -2,0,-2: DRAW -4,4: DR
AW -12,-3,2
  75 DRAW -5,-6,-1: DRAW 0,-5,3:
 DRAW 4,-2,-2: DRAW 3,0,2: DRAW
3,9
  80 DRAW 2,-2,-2: DRAW 0,-5,2:
DRAW 0,-2,1: DRAW -2,-4,2: DRAW
-8,-1
  85 DRAW -1,4,-1: DRAW -2,0,3:
DRAW -8,-8,-2
  90 DRAW 0,-1: DRAW 2,-3,-3
  95 DRAW -3,0: DRAW -1,-1: DRAW
 0,-5: DRAW 3,-1,-1: DRAW 3,4,2:
 DRAW 5,3,-1
 100 DRAW 5,-3: DRAW 2,-2,3: DRA
W 1,2: DRAW -4,4
```



```
 105 DRAW 2,2,-3: DRAW 4,-4: DRA
W 1,-3: DRAW 2,0,2: DRAW 1,3,-2
 110 DRAW 4,2: DRAW 1,4,-1: DRAW
 4,-2,3: DRAW 0,-3,-3: DRAW -6,-
2,2: DRAW 0,-3,3
 115 DRAW 0,-5,-3: DRAW -11,-1,1
: DRAW -4,5,-1: DRAW -10,-4,.5
 120 DRAW -4,-21,2: DRAW 10,0,1:
 DRAW 5,-3,-3: DRAW 0,-16,-1: DR
AW 5,-13: DRAW 7,0,2
 125 DRAW 4,10,1: DRAW 3,12,2: D
RAW 6,10,-1: DRAW -3,3,3: DRAW -
7,7,-2: DRAW -2,7
 130 DRAW 2,0,-3: DRAW 6,-11: DR
AW 3,-1,3: DRAW 7,4: DRAW -1,3,3
: DRAW -5,4,-1: DRAW 2,1,-3: DRA
W 8,-3,1: DRAW 8,-12,-1: DRAW 4,
0,2
 135 DRAW 6,10,-1: DRAW 2,-5: DR
AW 0,2: DRAW 5,-12,1: DRAW 2,0,3
: DRAW -3,8,-1: DRAW 4,-4,-1: DR
AW 1,10,2: DRAW 6,15,1
 140 DRAW 5,-4,-1: DRAW 2,0,3: D
RAW 0,4,-1: DRAW 5,13,1: DRAW -3
,4: DRAW 7,6,-2: DRAW 6,2,-1: DR
AW -1,-9,1: DRAW 2,0,3: DRAW 4,1
0,.5
 145 DRAW 10,4,1
 150 PLOT 154,70: DRAW 0,8,-.5:
DRAW 3,6,-2: DRAW 0,-6,-.5: DRAW
 -3,-8,-2
 160 PLOT 216,60: DRAW -7,5,-2:
DRAW -4,1,1: DRAW -6,-3,2: DRAW
-4,2,-3
 165 DRAW 10,15,-2: DRAW 4,0,-4:
 DRAW 3,-1,3: DRAW 1,3: DRAW 2,0
,-2: DRAW 6,-9,2: DRAW -5,-14,-2
 170 PLOT 105,150: DRAW 7,-1,3:
DRAW -2,2,2: DRAW -5,-1,1
 175 PLOT 209,120: DRAW 7,4,1: D
RAW -2,2,3: DRAW -6,-4,-1: DRAW
0,-2,3
 180 PLOT 47,159: DRAW 3,-2,-2:
DRAW -2,-2,-1: DRAW -2,2,-2: DRA
W 1,2,-2
 185 PLOT 212,85: DRAW -4,5,-2:
DRAW 7,0,-1: DRAW 4,-3,1.5: DRAW
 1,-3,-4: DRAW -7,1,1
 190 PLOT 156,164: DRAW 2,-2,-3:
 DRAW -4,-5,2: DRAW -2,3,-3: DRA
W 3,4,-1
 195 PLOT 179,98: DRAW -1,-3,-2:
 DRAW 1,3,-3
```

PROGRAMA 4

```
200 PLOT 88,148: DRAW -2,12,-1: DRAW -2,2,2: DRAW -5,2,-1.5: DRAW 22,6,-2: DRAW 4,-1,3: DRAW 3,-2,-2: DRAW -1,-8,-1: DRAW -5,-6,-1: DRAW -5,-5,1.5: DRAW -9,0,-3
205 PLOT 226,53: DRAW 5,5,-1: DRAW -5,-5,-2: PLOT 234,57: DRAW 1,4: DRAW 3,-1,-2: DRAW -3,-3,-1.5
210 PLOT 66,108: DRAW 3,0,-2: DRAW 3,-2,-2: DRAW -6,2: PLOT 68,108: PLOT 71,107: PLOT 74,104: DRAW 3,2,2: DRAW -3,-2,2
215 PLOT 83,150: DRAW -10,9,1: DRAW -5,-2,2: DRAW 3,-2,2: DRAW 2,-7,-2: DRAW 5,-3,-1: DRAW 6,4,2
220 PLOT 217,133: DRAW -2,-5,1: DRAW 3,0,2: DRAW -1,5,1.5
225 PLOT 117,136: DRAW 3,-1,2: DRAW 0,4,2: DRAW -2,4,-1: DRAW -2,-1,3: DRAW 1,-6,-1
230 PLOT 112,136: DRAW 2,3,2: DRAW -2,-3,2
235 PLOT 191,81: DRAW -8,10,1: DRAW 8,-10,1: PLOT 197,88: DRAW 1,6,2: DRAW -3,-2,-1: DRAW 2,-4,2
240 PLOT 72,173: DRAW 10,1: DRAW -2,-3,-3: DRAW -7,-3,1: DRAW -3,1,-3: DRAW 0,3,3: DRAW 2,1,-4
245 PLOT 55,155: DRAW 6,1,2: DRAW -9,3,2: DRAW 3,-4,2
250 PLOT 142,113: DRAW -1,2: PLOT 152,132: DRAW 0,18
255 PAUSE 20: FOR n=22528 TO 23231: POKE n,45: NEXT n
260 PRINT AT 17,0; PAPER 0,,
265 FOR n=18 TO 21: PRINT AT n,0; PAPER 7,,: NEXT n
270 PRINT INK 0;AT 0,0;"K";AT 0,31;"L";AT 16,0;"M";AT 16,31;"N"; PAPER 7;AT 18,0;"K";AT 18,31;"L";AT 21,0;"M";AT 21,31;"N"
275 PRINT INK 0;AT 14,1;"U"; INK 1;"O"; INK 0;"T";AT 13,2;"^";AT 15,2;"S";AT 12,2; INK 1;"N"
280 POKE 22638,46: POKE 23050,44: POKE 22962,43
285 LET cm=1: GO SUB 1000: GO SUB 3500: GO SUB 9000
```

```
500 REM ▓▓▓Calavera▓▓▓
505 REM
510 PAPER 0: INK 7: BORDER 0: CLS
515 PLOT 90,80: DRAW -20,50,-1
520 DRAW 100,0,-2.5: DRAW -20,-50,-1
530 DRAW -5,-5,1: DRAW 0,-20: DRAW -50,0,-3
540 DRAW 0,20: DRAW -5,5,1
550 CIRCLE 95,120,15: CIRCLE 145,120,15
560 PLOT 120,100: DRAW 9,-12,-1: DRAW -18,0,-1: DRAW 9,12,-1
570 PLOT 95,70: DRAW 50,0: PLOT 95,54: DRAW 50,0
580 FOR n=95 TO 144 STEP 10
590 PLOT n,70: DRAW 0,-6: DRAW 10,0,1: PLOT n,54: DRAW 0,6: DRAW 10,0,-1
600 NEXT n
605 PLOT 81,89: DRAW -61,31: PL
```



```
OT 95,70: DRAW -80,40
 610 PLOT 145,57: DRAW 75,-37: P
LOT 143,46: DRAW 72,-36
 615 PLOT 95,57: DRAW -75,-37: P
LOT 95,46: DRAW -72,-36
 620 PLOT 159,89: DRAW 61,31: PL
OT 145,70: DRAW 80,40
 630 RESTORE 1585: GO SUB 1010
 640 POKE 58445,76: POKE 58446,2
04: POKE 58452,24
 650 RANDOMIZE USR 58444: LET cm
=0: RESTORE 9035: GO SUB 9015: G
O TO 9500
1000 REM ▓▓▓ Relleno ▓▓▓
1005 REM
1007 RESTORE 1000
1010 READ yi,xi: IF yi>=1000 THE
N  INK yi/1000: GO TO 1010
1015 IF yi=999 THEN  GO TO 1075
1020 FOR s=-1 TO 1 STEP 2: LET x
is=xi-(s<0): LET xs=300*s
1025 FOR n=yi TO 175
```



```
1030 FOR m=xis TO xs STEP s
1040 IF POINT (m,n)=1 THEN  GO T
O 1060+5*(m=xis)
1050 PLOT m,n
1055 NEXT m
1060 NEXT n
1065 NEXT s
1070 GO TO 1010
1075 RETURN
1500 REM ▓▓ Datos de relleno ▓▓
1505 REM
1510 DATA 1000,0,98,64,96,68,105
,55,135,17,134,17,150,40,150,21,
108,45
1515 DATA 110,45,111,43,113,42,1
51,58,134,73,151,66,152,68,113,6
7,114,66
1520 DATA 115,65,105,75,146,79,1
57,72,155,57,156,48,157,47,145,9
2,168,107,168,72,173,80
1525 DATA 2000,0,97,173,100,151,
126,145,153,153,154,161,156,166,
161,183,158,192,155,205
1530 DATA 153,225,96,178,118,204
,120,204,138,221,100,190,97,191,
95,186,93,188,94,187,97,185,100,
184
1535 DATA 121,210,128,216,132,21
7,122,212,123,214,158,154,161,15
4,163,156
1540 DATA 1000,0,60,217,61,214,6
3,197,78,207,80,214,86,213
1545 DATA 85,221,54,229,58,236,5
7,230,83,189,84,189,87,187,89,18
5,89,196,92,198
1560 DATA 4000,0,44,77,3000,0,62
,137,115,126,95,118,71,155,73,15
4,79,154,83,156
1565 DATA 6000,0,126,139,139,131
,134,125,120,117,123,138,126,121
,136,118
1570 DATA 142,117,122,131,124,13
0,125,128,126,127,137,113,147,10
9
1580 DATA 999,0
1585 DATA 33,120,71,120,101,120
1590 DATA 105,78,105,162,86,95,8
6,145
1595 DATA 999,0
1600 DATA 9999
2000 REM ▓▓▓ Verif. Datos ▓▓▓
2005 REM
2010 PRINT "Verificando Datos."
```

```
2015 RESTORE 1500: LET s=0
2020 READ a: LET s=s+a: IF a=999
9 THEN  GO TO 2030
2025 GO TO 2020
2030 IF s<>54715 THEN  PRINT "Er
ror en datos de relleno.": STOP
2035 RESTORE 9000: LET s=0: FOR
n=1 TO 24: READ a: LET s=s+a: NE
XT n: IF s<>2144 THEN  PRINT "Er
ror en datos de c. maquina.": ST
OP
2040 RESTORE 3000: LET s=0: FOR
n=1 TO 192: READ a: LET s=s+a: N
EXT n: IF s<>15484 THEN  PRINT "
Error en datos de graficos.": ST
OP
2050 PRINT "Datos Correctos.": R
ETURN
3000 REM ▓▓▓ Graficos ▓▓▓
3005 REM
```

```
3040 DATA 63,15,3,3,1,1,0,0,0,0,
128,128,192,192,240,252,0,0,1,1,
3,3
3045 DATA 15,63,36,131,42,65,2,1
60,44,81,0,0,0,254,254,56,16,16,
0,0,254,254,56,16,16,56
3050 DATA 0,0,254,254,124,124,16
,56,0,16,16,16,16,84,56,16,0,4
3055 DATA 2,127,2,4,0,0,0,32,64,
254,64,32,0,0
3500 RESTORE 3500: FOR n=65512 T
O 65535: READ a: POKE n,a: NEXT
n: RETURN
3505 DATA 84,56,254,56,84,16,0,0
,124,238,198,238,124,56,0,0,84,1
24,84,56,0,0,0,0
9000 REM ▓▓▓ Codigo maquina ▓▓▓
9005 REM
9010 RESTORE 9000
9015 FOR n=58444 TO 58455: READ
a: POKE n,a: NEXT n
```




```
3010 RESTORE 3000: FOR n=0 TO 16
7: READ a: POKE 65368+n,a: NEXT
n
3015 RETURN
3020 DATA 243,226,192,17,192,226
,243,255,255,247,227,227,128,0,1
31,195
3025 DATA 255,193,1,231,0,68,129
,255,255,239,199,199,199,239,199
,187,0,0,0,24,24,0,0,0,0,6,6,0
3030 DATA 0,96,96,0,0,6,6,24,24,
96,96,0,0,102,102,0,0,102,102,0,
0,102
3035 DATA 102,24,24,102,102,0,0,
102,102,66,36,102,102,0,252,240,
192,192,128,128,0,0
```

```
9020 IF cm=1 THEN  RANDOMIZE USR
 58444
9025 RETURN
9030 DATA 17,88,228,33,0,64,1,0,
27,237,176,201
9035 DATA 17,0,64,33,88,228,1,0,
27,237,176,201
9500 REM  ▓▓▓ Grabacion  ▓▓▓
9505 REM
9510 PAUSE 100: INK 0: PAPER 7:
CLS : PRINT "Grabacion."
9515 SAVE "TLBcm"CODE 52300,1323
5
9525 INPUT "Mas copias? (s/n) ";
 LINE a$: IF a$="s" THEN  GO TO
9515
```

Preguntas y respuestas

P Quería felicitaros por el artículo publicado en el número 10, titulado «Protección del software», y quería ayudar a completarlo. En la página 15 se dice: «... Recientemente han aparecido varios programas que parece ser han conseguido desterrrar del reino del Spectrum a los más potentes copiadores. Son los llamados programas TURBO, grabados a distinta velocidad del estándar prefijado por las rutinas de la ROM». Pues bien, tengo en mis manos las instrucciones de un programa copiador en TURBO, que además copia programas normales. El programa se llama Lerm 6, y sus especificaciones técnicas son las siguientes: carga normal, carga con cabeceras cortas, carga con cabeceras muy largas, grabación con cabecera ancha, grabación normal, grabación hasta 41 K, grabación en bloques inferiores a 47,3 K, grabación de bloques hasta 50 K, realiza copias tanto en cassette como en microdrive, graba programas sin carga TURBO, carga programas turbo en 9 velocidades, pudiéndose alterar cuando se salve su velocidad, hasta 50 K.

Gabriel Pérez
León

R *Efectivamente la carrera entre piratas y programadores no ha hecho más que comenzar. Constantemente surgen nuevos métodos de protección, cada día más complejos. Estos apenas dan a las grandes compañías un pequeño respiro que deben aprovechar para sacarle el máximo jugo a sus últimos programas. Pero, ningún método es inexpugnable, y la piratería se ha convertido en el deporte preferido de cientos de programadores, que no tardan mucho en dar con la rutina correcta que eche por tierra la protección de moda. Surgen programas piratas que son capaces de copiar a otros programas piratas, otros que se copian a sí mismos. En ocasiones resulta más rentable para las casas de soft el comercializar programas que esquiven las protecciones de sus últimos éxitos, antes de que lo hagan otras compañías y sean ellas las que se lleven los beneficios. A este paso no andará lejos el día en que se vendan más copiadores piratas que juegos y utilidades. De hecho, ya ha caído en nuestras manos el número 7 de la serie a que haces alusión (Lerm 7), que es, como podrás imaginar, una versión modernizada del programa que comentas.*

P Soy un coleccionista de programas deportivos para el Spectrum. En mi colección falta algún programa de fútbol, de los cuales no encuentro en ninguna parte. ¿Me podrían decir si hay programas de fútbol para Spectrum?, y si los hay, ¿qué casa los comercializa de mejor calidad?

Enrique Ramírez
Marbella (Málaga)

R *Entre los programas de fútbol tradicionales podemos destacar «Match day», de Ocean, y «World Cup», de Artic, ambos bastante logrados (mejor el primero) y para Spectrum 48 K. Si te gusta el fútbol (rugby) americano y conoces su reglamento, también podrás disfrutar con «American football», de Mind Games. Y si prefieres ver los partidos desde el banquillo, lo tuyo es el «football manager», distribuido por Investrónica, en el que deberás llevar las finanzas, fichajes, etc., de un equipo a lo largo de una o varias ligas. Sea cual sea tu elección, esperamos que no te «cuelen» muchos goles.*

P Soy un asiduo lector de vuestra revista y quisiera, ante todo, felicitaros por los buenos artículos publicados en ella.

Mi problema es el siguiente: He copiado el programa de ampliación del sistema operativo publicado en el número de septiembre, y creo que hay un error en la primera línea, pues al ejecutarla siempre me da error en dicha línea. Ruego publiquéis esta errata. Gracias. Seguid publicando buenos programas de utilidades como éste.

Eduardo Maya
Lanjarón (Granada)

R *Dadas las características de la línea a que haces alusión, y suponiendo que las hayas copiado co-*



rrectamente, el error sólo puede residir en la primera de sus sentencias. Esta hace un CLEAR para bajar RAMTOP y dejar así un espacio de memoria disponible para colocar allí el código máquina. Esto está pensado, y en el artículo quedaba bien claro, para un Spectrum 48 K, por lo que si el tuyo es un 16 K, el ordenador nunca podrá ejecutarlo, ya que el comando se referiría a una posición de memoria no existente.

Si no es este tu problema, sólo podemos aconsejarte que leas atentamente las instrucciones de carga del programa y que, caso de continuar los problemas, nos escribas una segunda carta especificando mejor las características de tu problema (diciéndonos, por ejemplo, el mensaje de error que da la máquina). El programa en cuestión, como todos los que publicamos, ha sido convenientemente probado y podemos asegurarte que carece de erratas.

Intentamos publicar la mayor cantidad posible de utilidades, ya que, a pesar de ser menos «comercializables» que los juegos, son muchos los aficionados como Manuel Arana que les dedican su tiempo.

P En el montaje del Interfacer Joystick que publicásteis en el número de julio se muestra en los dibujos correspondientes un circuito marcado con el nombre 74SL144. Pues bien, si se consulta la lista de componentes de dicho montaje, se dice que el nombre de este circuito es 74LS244. ¿Cuál es el que tenemos que comprar los que queramos realizar este montaje?

Juan J. García
León

P Me dirijo a Vds. para rogarles me aclaren el artículo del Interface Joystick.

1) Pocas especificaciones para una cosa tan delicada en el sentido de que te cargas el ordenador.

2) La fotografía de la página 28 no corresponde con el dibujo de la placa.

3) No sé si es un circuito de doble cara o simple, pero tiene que ser doble por que no coje de otra manera. La foto es circuito simple o no es el mismo. Si es doble la patilla del 74LS32 está a 112 V, la 7 que es masa está libre, la 20 del 74LS144 (o 74LS244, que no se sabe cuál) está a D7 en vez de a +5 V. ¿Por favor, pueden aclararme esto?, pues tengo hecha la placa y se me ocurrió probar antes un poco y no hay forma. Imagino que será el 74LS244 o el 240, porque el 144 no lo conozco.

P. Horcasada
Valencia

R *La fotografía de la página 28 efectivamente no debe tenerse en cuenta, ya que corresponde a un montaje de interface preliminar y está puesta a modo de indicación para un montaje del conector en posición ventral.*

El circuito es doble cara. El chip del buffer es el 74LS244. Ciertamente falta una pequeña pero importantísima pista: Ha de ponerse a masa la patilla 7 del chip 74LS232. Y por último, el 74LS244 tiene alimentacón Vcc a 5+ en la patilla 20 y masa en la patilla 10, como está correctamente impreso en la figura.

P Llevé mi ordenador (un Spectrum 3B) a reparar al servicio técnico «Hissa» de Vigo, y cuando lo puse a funcionar me di cuenta de que el programa «Boxeo» del número 2 no me funcionaba. Verifiqué la versión que era con PRINT IN 63486 y obtuve 191 como respuesta. Cuando pruebo PRINT IN 65022 también obtengo como respuesta 191, y según vi en una revista, mi versión tendría que ser 3B o Plus. ¿Es posible que me hayan cambiado a 3 la versión?

José Luis González
Sta. Cristina de Cobres (Pontevedra)

El corcho

Vendo Commodore 16 nuevo y sin estrenar con manuales en inglés y castellano. Precio entre 30.000 y 25.000 pesetas. Llamar los interesados al teléfono: (91) 473 70 61 de 2 a 3 tarde ó 9 a 11 noche, preguntar por Luis.

Vendo gran cantidad de circuitos integrados, para ordenador (micros, memorias, periféricos) y demás usos generales. Mando lista sin compromiso. Interesados dirigirse a: Miguel Angel Clavijo. José Antonio, 5. 1.º izda. La Cuesta. Santa Cruz de Tenerife.

Vendo Spectrum 48 K completo con garantía y cassette más de 100 programas últimas novedades por 40.000 pesetas. También vendo unidad de disco «Discovery 1» en 55.000 pesetas con garantía. Llamar al teléfono (985) 29 21 87.

Vendo Wafadrive en esplendidas condiciones (comprado en julio-85), aun con garantía de 6 meses. Regalo tres Wafers (microcintas), procesador de textos Spectral Writer (en inglés y otro traducido por mí al castellano), con copia de seguridad en cinta; manuales en ingles y facturas. Solicita cuanta información precises. Todo sólo por 30.000 pesetas. Aprovecha esta ocasión. Llamar al teléfono (981) 78 29 52 (20 h.). José Manuel Cagiao. Apartado 2.144 de La Coruña (15080).

Su anuncio puede ir aquí. Escríbanos a «El corcho». TODOSPECTRUM. Bravo Murillo, 377. 5.º A. 28020 Madrid.

Preguntas y respuestas

P *Es totalmente lógico que al haberte cambiado piezas defectuosas, y más aún, si una de éstas ha sido la ULA, haya cambiado la versión de tu Spectrum; ya que las tan nombradas versiones se diferencia en sólo algunos detalles. Es posible, por tanto, que tengas ahora una versión mixta, o incluso que, al realizar la reparación, sustituyeran la totalidad de la placa de circuito impreso. De todas formas esto no debe influir en el funcionamiento de tu ordenador; deberás, eso sí, efectuar las modificaciones necesarias en los programas que utilicen sentencias IN para explorar el teclado.*

R Me gustaría que me informaran de las características técnicas y representante en España de la impresora Canon A-1210.

Miguel Agulló
San Antonio (Ibiza)

P *Sus características son las siguientes:*

— Impresión por inyección de tinta.

— Color.

— Bidireccional.

— Número máximo de caracteres por línea: 40 (elongados), 80 (normales).

— Líneas por pulgada: 6.

*— Dimensión de la matriz: 5*7 puntos.*

— Juego de caracteres: ASCII (128 caracteres).

— Velocidad: 40 caracteres por segundo.

— Papel: normal, en rollo, anchura máxima: 8,5".

— Número de copias: 1.

— Tipo de arrastre: fricción.

— Interfaces: paralelo Centronics estándar, serie RS-232.

— Buffer: 1 línea (40-80 caracteres).

*— Resolución: 560*4 colores (gráfica), 650*4 colores (copia TRC).*

— Nivel de ruido: 85 dB.

— Representante en España: Canon Copiadoras de España, S. A. Avda. Menéndez Pelayo, 67. Madrid. Tel.: (91) 273 75 08.

R Somos un grupo de amigos con varios ordenadores y quisiéramos que nos resolvieseis algunas dudas:

¿Qué ventajas e inconvenientes reales tiene la ampliación interna de memoria del Spectrum? Ante la posibilidad de adquirirla fuera de España, ¿tiene su montaje algún inconveniente para realizarlo uno mismo? ¿Qué pasa con la garantía del ordenador?

Entre la publicidad de vuestra revista hemos visto que existe la posibilidad de conectar varios Spectrum formando una «Red» que permitirá así «compartir programas» entre ellos. ¿Esta Red equivaldrá a una ampliación de memoria RAM y, por tanto, permitirá introducir en el ordenador programas de más de 48 K?

Victoria Arauzo
y Eduardo Nogueira
Madrid

R *La ampliación de memoria interna, caso de estar correctamente hecha, convertirá a tu Spectrum en un 48 K sin ningún tipo de limitaciones. Si la adquirís en el extranjero deberéis cuidar que quien realice el montaje tenga alguna experiencia en la materia, pues os jugáis el ordenador. La garantía queda invalidada una vez se rompan los precintos que cubren los tornillos inferiores.*

El hecho de que un ordenador pueda acceder a la memoria de sus compañeros de red no es sinónimo de ampliación de memoria en el habitual sentido que esta frase tiene. Esto no quiere decir que no puedan usarse estas Ks extras para almacenamiento de datos, pero un microprocesador nunca podrá, por medio de la red, «correr» un programa en la memoria de otro ordenador.

DIRECTOR:
Fernando García
COORDINADOR EDITORIAL:
Emiliano Juárez
REDACCION:
Enrique Larreta
Juan Arencibia, Fernando García, José C. Tomás, Luis M. Brugarolas. Santiago Gala
DISEÑO: Ricardo Segura

Editado por
PUBLINFORMATICA, S. A.
Presidente: Fernando Bolín

Administración:
INFODIS, S. A.
Gerente de Circulación y ventas:
Luis Carrero
Producción:
Miguel Onieva
Director de Marketing:
Antonio González
Servicio al cliente:
Julia González. Tel. 733 79 69
Administración:
Miguel Atance
Jefe de Publicidad
María José Martín
Dirección y redacción:
Bravo Murillo, 377-5.º A. Tel. 733 74 13
Telex: 48877 OPZX e 28020 Madrid
Administración y Publicidad:
Bravo Murillo, 377-3 E. Tels. 733 96 62/96
Publicidad Madrid:
María Jose Martín
Publicidad Barcelona:
María del Carmen Ríos, Olga Martorell. Pelayo, 12.
Tel. (93) 318 02 89.
08001 Barcelona.
Depósito legal: M-29041-1984
Distribuye S.G.E.L.
Avda. Valdelaparra, s/n.
Alcobendas-Madrid.
Fotomecánica: Karmat, C/ Pantoja, 10. Madrid.
Fotocomposición: Artecomp.
Imprime: Héroes, C/ Torrelara, 8. Madrid.

Distribuidor en VENEZUELA,
SIPAM, S.A.
AVD. REPUBLICA DOMINICANA, EDIF. FELTREC – OFICINA 4B
BOLEITA SUR
CARACAS (VENEZUELA)
Esta publicación es miembro de la Asociación de Revistas de Información ARI asociada a la Federación Internacional de Prensa Periódica, FIPP.

SUSCRIPCIONES:
Rogamos dirijan toda la correspondencia relacionada con suscripciones a:
TODOSPECTRUM
EDISA: Tel. 415 97 12
C/ López de Hoyos, 141-5.º
28002 MADRID
(Para todos los pagos reseñar solamente TODOSPECTRUM)
Para la compra de ejemplares atrasados diríjanse a la propia editorial
TODOSPECTRUM
C/ Bravo Murillo, 377-5.º A
Tel. 733 74 13-28020 MADRID

Si deseas colaborar en TODOSPECTRUM remite tus artículos o programas a Bravo Murillo 377, 5.º A. 28020 Madrid. Los programas deberán estar grabados en cassette y los artículos mecanografiados.
A efectos de remuneración, se analiza cada colaboración aisladamente, estudiando su complejidad y calidad.

ORDENADOR PERSONAL

Canon

V-20

SISTEMA MSX

Y ADEMAS PUEDE VIAJAR A MEJICO CON LA SELECCION PREGUNTE A SU PROVEEDOR

DISFRUTELO EN FAMILIA

El ordenador personal para toda la familia. Con 80 K para estar a la altura de todos los gustos y necesidades. Jugar, archivar, aprender, programar: y con capacidad para crecer con la aplicación de periféricos.

CARACTERISTICAS MAS IMPORTANTES DEL V-20

- Sistema standard MSX que hace compatibles hardware y software de todos los productos que tienen este sistema en el mercado.
- Pueden acoplar los siguientes periféricos de CANON:
  - — Impresoras.
  - — Floppy de 720 K, que incluye diskette MSX-DOS para aprovechar toda la capacidad del ordenador y además incluimos un segundo diskette con tres programas de aplicaciones profesionales.
  - — Mouse para hacer todo tipo de gráficos a color.
  - — Joy sticks.
  - — Caja de 5 diskettes vírgenes.
  - — Variedad de programas en cinta con juegos.
- Y además dos manuales en castellano: guía del usuario y completo manual de BASIC.